A Unified Theory of Representation Learning: How Hidden Relationships Power Algorithms that can Learn without Labels

by

Mark T. Hamilton

B.S., Yale University (2016)
M.S., Massachusetts Institute of Technology (2022)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

DOCTORATE OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

Authored by:	Mark T. Hamilton Department of Electrical Engineering and Computer Science May 16, 2025
Certified by:	William T. Freeman Thomas and Gerd Perkins Professor of EECS Thesis Supervisor
Accepted by:	Leslie A. Kolodziejski Professor of Electrical Engineering and Computer Science Chair, Department Committee on Graduate Students

THESIS COMMITTEE

Thesis Supervisor

William T. Freeman
Thomas and Gerd Perkins Professor of EECS
Department of Electrical Engineering and Computer Science

Thesis Readers

Vincent Sitzmann
Assistant Professor of Computing
Department of Electrical Engineering and Computer Science

Phillip Isola
Associate Professor of Computing
Department of Electrical Engineering and Computer Science

For Lily, Kerry, Mom, and Dad.

A Unified Theory of Representation Learning: How Hidden Relationships Power Algorithms that can Learn without Labels

by

Mark T. Hamilton

Submitted to the Department of Electrical Engineering and Computer Science
on May 16, 2025 in partial fulfillment of the requirements for the degree of

DOCTORATE OF PHILOSOPHY

ABSTRACT

How does the human mind make sense of raw information without being taught how to see or hear? This thesis presents a unifying theory that describes how algorithms can learn and discover structure in complex systems, like natural images, audio, language, and video - *without human input.* This class of algorithms has the possibility to extend our own understanding of the world by helping us to see previously unseen patterns in nature and science. At the core of this thesis' unified theory is the notion that *relationships* between deep network representations hold the key discover the structure of the world without human input. This work will begin with a few examples of this principle in action; discovering hidden connections that span cultures and millennia in the visual arts, discovering visual objects in large image corpora, classifying every pixel of our visual world, and rediscovering the meaning of words from raw audio, all without human labels. In the latter half of this thesis, we will present two unifying mathematical theories of unsupervised learning. The first will explain why relationships between deep features can rediscover the semantic structure of the natural world by connecting model explainability, cooperative game theory, and deep feature relationships. The second mathematical theory will show that relationships between representations can be used to unify over 20 common machine learning algorithms spanning 100 years of progress in the field of machine learning. In particular, we introduce a single equation that unifies classification, regression, large language modeling, dimensionality reduction, clustering, contrastive learning, and spectral methods. This thesis uses this unified equation as the basis for a "periodic table of representation learning" that predicts the existence of new types of algorithms. We show that one of these predicted algorithms is a state-of-the-art unsupervised image classification technique. Finally, this work will summarize the key findings and share ongoing and future directions,.

Thesis supervisor: William T. Freeman
Title: Thomas and Gerd Perkins Professor of EECS

Acknowledgments

Finishing a Ph.D. is impossible without the support, insight, care, and kindness of an extraordinary network of individuals. I am deeply grateful to all those who have helped me along the way.

To my advisor Bill, thank you for turning what many describe as a grueling process into one of the most fulfilling and joyful periods of my life. Your decades of experience, high-level insight, and genuine care have been an inspiration to me. From your example, I have learned not just how to conduct research, but also how one should lead a research group. You have provided me with a safe and encouraging place to study anything that my heart desires and have supported me with resources, feedback, and help. Thanks for all the amazing conversations, laughs, and fundamentally changing the direction of my life. I am incredibly lucky to have had your mentorship.

To my committee members, Vincent and Phillip, your collective brilliance has long been an inspiration to me, and I have blatantly copied your playbooks for several of the papers in this thesis. My conversations with you have been pivotal in helping me grow as a researcher, and your wisdom has been instrumental in guiding the direction of my work. I am grateful for their time (especially to read a thesis!), patience, and invaluable feedback.

I owe a huge amount to the mentors who have shaped my life. Thanks to Herb Weiss who first introduced me to scientific research in high school, helping me make the step to college, connecting me with professors for summer research. Thanks to Sajan Saini for the kindness and patience during my first scientific research project. I will never forget how patiently he taught me vector calculus and electrodynamics. Thanks to Meg Urry, Uri Shaham, and Sahand Negahban, for their support and guidance at Yale as I entered the field of machine learning. Thanks to Sudarshan Raghunathan, my first manager at Microsoft, for showing me what caring for employees really means. He not only taught me how to code, but taught me what it means to own and drive a project. I am grateful for his support when I decided to apply for PhD programs and even more grateful for his help setting up the MIT/MSFT split. Thanks to Anand Raman, who met with me at times more frequently than any of my managers. Anand taught me to dream big and think outside the box when it comes to releasing a project. Both he and Sudarshan helped keep SynapseML alive and were constantly championing my work. Thank you to Joseph Sirosh, Markus Cozowicz, and Wee Hyong Tok for their friendship, mentorship, and providing the opportunity to share my work on stages that I could not have imagined. Thank you to my Ph.D. letter writers Sudarshan, Anand, Joseph, Uri, and Adam Kalai for helping me start this phase of my life. I am thankful for my mentors during my Ph.D. including Andrew Zisserman, Simon Stent, John Hershey, and Scott Lundberg. Our technical discussions changed the way I thought about almost everything in deep learning. Thanks to them for sharing some of their genius. Finally, thanks to Markus Weimer and Avrilia Floratau for providing me with a supportive environment to push on both MSFT and MIT projects, and thanks especially for the care, mentorship, and support they both showed me during my own journey to become a manager. Without them, SynapseML would have never made it to general availability.

Behind the scenes, countless others made this journey possible. Thanks to Roger White who keeps the lab running smoothly for helping organize my defense, celebrations, travel. Joel Emer, for his academic wisdom and camaraderie. Janet Fisher and Leslie Kolodziejski, who have kept the graduate program running smoothly. To Rachel Gordon, Alex Shipps, and Adam Zewe from the MIT and CSAIL media teams. Your work has amplified my research in ways I could never have imagined. These days, the impact of a paper is 40% content and 60% presentation, and you have certainly made sure the presentation half was world-class. Deep thanks go to the Microsoft Research Grand Central Resources team for their gracious help performing the experiments in this work. Special thanks to Oleg Losinets and Lifeng Li for their consistent, gracious, and timely help, debugging, and expertise. Without them, none of the experiments could have been run.

Bill's lab has been an intellectual home for me. The lab meetings, collaborations, and ideas the group shared kept my mind current, challenged, and inspired. To all my co-authors, thank you for the late nights, the review rebuttals, and the shared victories. Special thanks to Shaden Alshammari, a brilliant mathematician who spearheaded the creation of the periodic table of machine learning. I learned so much from you on this project and it is now the crown result of this thesis.

At Microsoft, I've been lucky to work with an outstanding engineering team. Thank you for trusting me as a manager during my PhD, and for keeping SynapseML moving forward. Thanks to Brendan Walsh, Jessica Wang, Farrukh Masud, Shyam Sai, Samhitha Mamindla, Niander Assis.

To my friends—thank you for the adventures and laughter. Whether we were skiing, sailing, cave diving, or just hanging out, your friendship has kept me grounded and joyful. I'm especially thankful to the many friendships formed during my time in Boston and at MIT. Thanks to Axel, Justin, AJ, Vasha, Anne, Theo, Evan, Stephanie, John, Logan, Chandler, Jiaqi, Emily, Sarah, Zhoutong, Praful, Kuan-Wei, and everyone I met at MIT for the fantastic conversations and memories. You've made this journey not just bearable, but truly wonderful.

To my family—thank you for being my constant foundation. I am blessed with a large, loving family that has always been there for me. To my cousins Mike, Danielle, Nicole, Kristen, and Holly, and my siblings Michelle, Steve, Joe, and Amy—thank you for your endless support. Thanks to Aunt Lo, who welcomed me into her house with open arms, we miss you dearly.

To Lily - my partner, my rock. Thank you for sharing this journey with me. Thank you for comforting me during rejections and celebrating with me during triumphs. I could not have asked for a better partner in crime during my Ph.D. You are my truffle hunter, my ski partner, my snorkel buddy, and my movie companion. You have taught me how to enjoy life to the fullest and have helped me stay healthy and happy during these challenging years. Your love, patience, and adventurous spirit have been a constant source of strength. I am so grateful for all the memories we have already made and so excited for the future we are building together.

To my sister Kerry brother Nick: Kerry, you have been a guide to me for as long as I can remember. From teaching me PEMDAS in middle school to co-authoring multiple papers with me during my Ph.D., your impact on my academic journey has been profound. You inspired me to try hard in school, to pursue science research, and to aim for the PhD. Nick, thank you for being a brother to me, and a pillar of support for my sister. We have all had a blast together on more trips than I can count, and looking forward to more. Congratulations on the arrival of baby Adelaide. She's lucky to have you as parents, and I can't wait to attend her dissertation soon.

To Mom and Dad—thank you for being the foundation of everything in my life. From the first days, you have been there with unwavering support, encouragement, and love. Dad, you've been my original problem solver, helping me build life-sized sharks in the garage and showing me how to break down big challenges with curiosity and determination. Mom, your consistent willingness to help, even if it means labeling thousands of illusory faces, is indicative of why you are so great. You read my most important drafts, support me no matter what, and are the biggest champion of my work. Together, you taught me the value of hard work, persistence, and dreaming big even when the path ahead was uncertain. You believed in me long before I believed in myself. I love you both and am endlessly grateful for everything you have given me.

Grant Acknowledgments: This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 2021323067. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

This research is based upon work supported in part by the Office of the Director of National Intelligence (Intelligence Advanced Research Projects Activity) via 2021-20111000006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

This work is supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/).

Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2- 1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

How does the human mind make sense of raw information without being taught how to see or hear? Humans have an extraordinary ability to learn about the world around them, often without explicit instructions or direct supervision. From infancy, we begin the almost *automatic* process of learning to recognize objects, interpret language, and navigate our environment mainly through self-directed observation and exploration. Our ability to convert raw sensory experiences into meaningful knowledge about the world helps us thrive in a dizzyingly complex world.

This “unsupervised” or “self-directed” way of learning is crucial not only for an individual, but also for humanity as a species. We collectively strive to advance scientific knowledge and make sense of the unknown. Over thousands of years, humans have voraciously devoured data in the name of discovering the secret order that underlies our natural world. For example, Dmitri Mendeleev discerned the periodic patterns of elements through careful study of chemical properties such as weight and reactivity. Similarly, Johannes Kepler uncovered the laws of planetary motion by analyzing astronomical data when no-one else believed that planets moved in elliptical orbits. In these cases, humanity’s scientists and thinkers were not taught the answers by their teachers; rather, they observed the natural world, studied the data, and used it to guide the formation of fundamentally new knowledge.

In stark contrast, traditional machine learning systems often rely on extensive datasets where each piece of data is explicitly labeled by human experts. These “supervised” methods, while powerful, are fundamentally limited by the quality of their labels. This limits our ability to use these kinds of algorithm when we don’t know what the correct answer should be. If we want to be able to automatically discover new science that rivals that of Kepler or Mendeleev, we must study algorithms that can go beyond what we already know and learn without direct human supervision.

This thesis will explore how we can create algorithms that can learn to see, hear, and understand the world directly from raw sensory data without human guidance. We will explore this problem from a variety of angles. We will see how these algorithms can rediscover many of the same concepts humans use in their daily lives like visual objects and language, and how they can give us *new* insights on existing datasets. This work will cover both the technical details behind specific systems and introduce deeper mathematical theories behind why they work. Towards the end of this work we will introduce a single equation that unifies more than 23 different commonly used machine learning methods in the broader literature and the thesis itself. We will use this equation as the basis for a new periodic table of machine learning and use the "gaps" in this table to predict new kinds of algorithms, just as Mendeleev used the "gaps" in his periodic table to predict the existence of new elements. We will see that the algorithms presented in earlier chapters appear as elements in this periodic table, and this lens will allow us to derive new algorithms that perform better than previously known methods. In this sense, this thesis will unify a broad swath of representation learning using a simple central idea: relationships between deep representations are the foundation upon which unsupervised algorithms are built.

More technically, each chapter of this thesis will study the ideas (commonly called "representations" or "features") learned by deep unsupervised learning algorithms. Although these algorithms never saw human labels, their representations display an emergent high-level semantic understanding of their training data. This thesis overcomes a key challenge: How can we extract useful and human-interpretable knowledge from the high-dimensional vectors commonly learned by deep networks? To this end, we show by considering the *relationships between* representations rather than the representations themselves, we can extract some of the rich information hidden in deep representations. In particular, each algorithm presented in this thesis shows that by studying, querying, and distilling these relationships, we can build systems that understand our world, convey this information clearly, and learn completely without human supervision. This observation is not just qualitative: we will see in Chapters 7 and 8 that this idea has the power to provide unifying theories for model explainability and representation learning as a whole.

1.2 Overview

Chapter	Brief Summary
2	Explains some key concepts that appear in the thesis in plain english. Good to read if you are in a field outside of ML.
3	"MosAIc" finds hidden cross-cultural and cross-media connections across large art museum collections using relationships between global visual representations.
4	"STEGO" discovers visual objects and classify every pixel of a visual dataset without labels using relationships between dense visual representations.
5	"FeatUp" upsamples any algorithm's dense visual representations by 64× by analyzing how these representations change as we jitter the algorithm's input. FeatUp shows that STEGO's discovered semantics can be made "pixel-perfect" without retraining.
6	"DenseAV" rediscovers the meaning of words in language and location of sound by analyzing the relationships between dense audio and visual representations.
7	We introduce an axiomatic theory for explaining the predictions of unsupervised models and search engines. These explanations are precisely the comparison between features used in DenseAV and STEGO.
8	We introduce a single equation (I-Con) unifying over twenty common learning methods and introduce a periodic table of representation learning. This equation is based on relationships between representations and generalizes both STEGO and DenseAV.

Table 1.1: Chapter summary and their connection to the main themes of this thesis: By studying the relationships between deep representations we can create algorithms capable of learning the structure of complex systems without human supervision.

This thesis will cover both the theory and practice of creating algorithms that learn without human guidance. The first few chapters will cover specific examples of these algorithms and how they can discover rich and intriguing structure from our natural world without human guidance. The latter half of this thesis will then formalize the ideas introduced in earlier chapters and show that they form the basis for a framework to unify 6 algorithms in model explainability (Chapter 7) and a single equation that unifies over 23 algorithms from the broader machine learning literature (Chapter 8). For quick overviews of the content of this thesis Table 1.1 presents 1 sentence chapter summaries, Figure 1.1 depicts the key ideas of this thesis in a graphical format, and Table 1.2. Chapter 2 provides simply written intuitive descriptions of key concepts that appear in the work for readers from other fields.

We begin this work with a first example of how relationships between deep-network features can help us find novel connections in the visual arts. Chapter 3 presents the MosAIc system, which finds hidden cross-cultural and cross-media relationships in the collected works of the Metropolitan Museum of Art and the Rijksmuseum. MosAIc discovers compelling visual analogies and artistic motifs that span both millennia and cultural barriers by studying the relationships between deep representations in large vision foundation models. This work not only helped thousands of museum visitors explore these art collections during the COVID-19 pandemic, but also produced a new technique that discovered "blind spots" in image generation algorithms where these methods systematically failed to capture the true diversity of their training data.

Chapter 3 shows that relationships between deep visual features capture important semantic similarities between objects, even if they appear in extremely varied context and settings. Chapter 4 builds on this idea and demonstrates that the very same pairwise affinities which let MosAIc spot kinship between artworks are also rich enough to define objects themselves. This chapter introduces STEGO, an algorithm that can discover a consistent ontology of visual objects and simultaneously classify every pixel of the world without human supervision anywhere in the pipeline. At its core, STEGO uses the relationships between *dense* deep visual features as its own form of "supervision" to train an unsupervised semantic segmentation system. This work doubled the performance of prior state-of-the-art methods and could rediscover the same objects as human annotators. Most interestingly, STEGO's purely unsupervised performance rivaled that of systems trained with thousands of human annotations across three diverse datasets: natural images, aerial land cover surveys, and first-person driving images. This work shows that large-scale visual categorization can emerge from analyzing relationships between dense visual representations, which later allowed the community to build methods for novel scientific domains such as medical imaging [1], LIDAR point clouds [2], animal behavior [3], and nanoelectronics [4].

Although chapter 4 shows that deep features can be used effectively for unsupervised semantic segmentation, this algorithm still suffered from the relatively low spatial resolution of many deep network representations. Chapter 5 tackles this long-standing practical obstacle: deep features are typically available only at coarse spatial resolutions (e.g. 7×7 for a ResNet-50), crippling algorithms like STEGO that use dense visual features to solve dense prediction tasks such as semantic segmentation, depth, or optical flow. FeatUp resolves this limitation by learning to up-sample deep network representations by up to 64× while preserving their exact semantics. Under the hood, FeatUp observes how a network's dense representations change as one varies the network's input image. By observing how slight changes to an image affect a network's deep representations, FeatUp can piece together the true high-resolution structure of the network's dense visual representations. FeatUp can up-sample any vision backbone without supervision and yields dramatic gains including sharper class-activation maps, +4–5 mIoU on semantic segmentation benchmarks, and clearer depth estimation. FeatUp therefore equips all subsequent chapters with rich semantic features with pixel-perfect resolution.

Equipped with the tools of Chapter 4 and 5, Chapter 6 extends the thesis from vision to multi-modal understanding. This chapter introduces DenseAV, an algorithm that uses relationships between deep audio and video features to rediscover the meaning of language without any human labels or text. DenseAV shows that by comparing deep feature representations, a network can rediscover the meaning of spoken words without ever seeing text or category labels. This finding generalizes the core hypothesis: meaningful abstractions arise from studying the relationships between deep features across modalities just as they do within a single sensory modality. It also serves as the start of a broader ongoing effort discussed in

Chapter 9 to create a machine learning system that can decode the communication of the Atlantic Spotted Dolphin.

Chapters 4 and 6 both show that the correspondences between dense visual features hold the secret to understanding language and visual objects without human supervision. Chapter 7 grounds this observation in mathematical theory and explains why and how these correspondences capture the semantics of the world so well. In particular, this chapter proves that the dense correspondences exploited by STEGO and DenseAV have a rigorous footing in cooperative game theory and model explainability. More formally, we interpret similarity-based predictions as games whose "players" are pixels or features. In this context, the Shapley and Harsanyi values of these games then assign each player an equitable share of the model's output. In this light, the dense feature similarities used in STEGO and DenseAV are precisely the "explanations" of the original unsupervised learner, explaining why they work so well in practice. Finally, we show this viewpoint generalizes and unifies several popular model explanation techniques such as Grad-CAM, Integrated Gradients, and LIME, and introduces efficient estimators that improve Harsanyi dividend convergence speeds by more than 10× compared to previous approaches.

Chapter 8 pushes further and explicitly formalizes the notion that relationships between representations are the key to unifying a broad swath of the machine learning literature. In particular, Chapter 8 introduces I-Con (Information-Contrastive), a single equation that unifies over 23 seemingly disparate learning objectives including classification, regression, LLMs, clustering, dimensionality reduction, contrastive learning, and spectral graph theory. This chapter shows that all of these approaches share the same underlying equation and can even be organized into a periodic table of representation learning. In this periodic table, every method can be viewed an example of distilling the relationships of one type of data-structure into another type of data-structure by minimizing a KL-divergence between two neighborhood distributions: one produced by a model, the other supplied (explicitly or implicitly) by data. We find that empty cells in our table predict the existence of novel algorithms. We show that one such a "gap in the periodic table" yields a state-of-the-art unsupervised image clusterer that improves ImageNet accuracy by eight percentage points. I-Con therefore elevates the thesis' central theme of studying the relationships between deep representations to the level of a universal equation, offering a road map for future discoveries beyond the works collected here.

Finally, Chapter 9 summarizes the key findings of this thesis and briefly describes ongoing research and future directions.

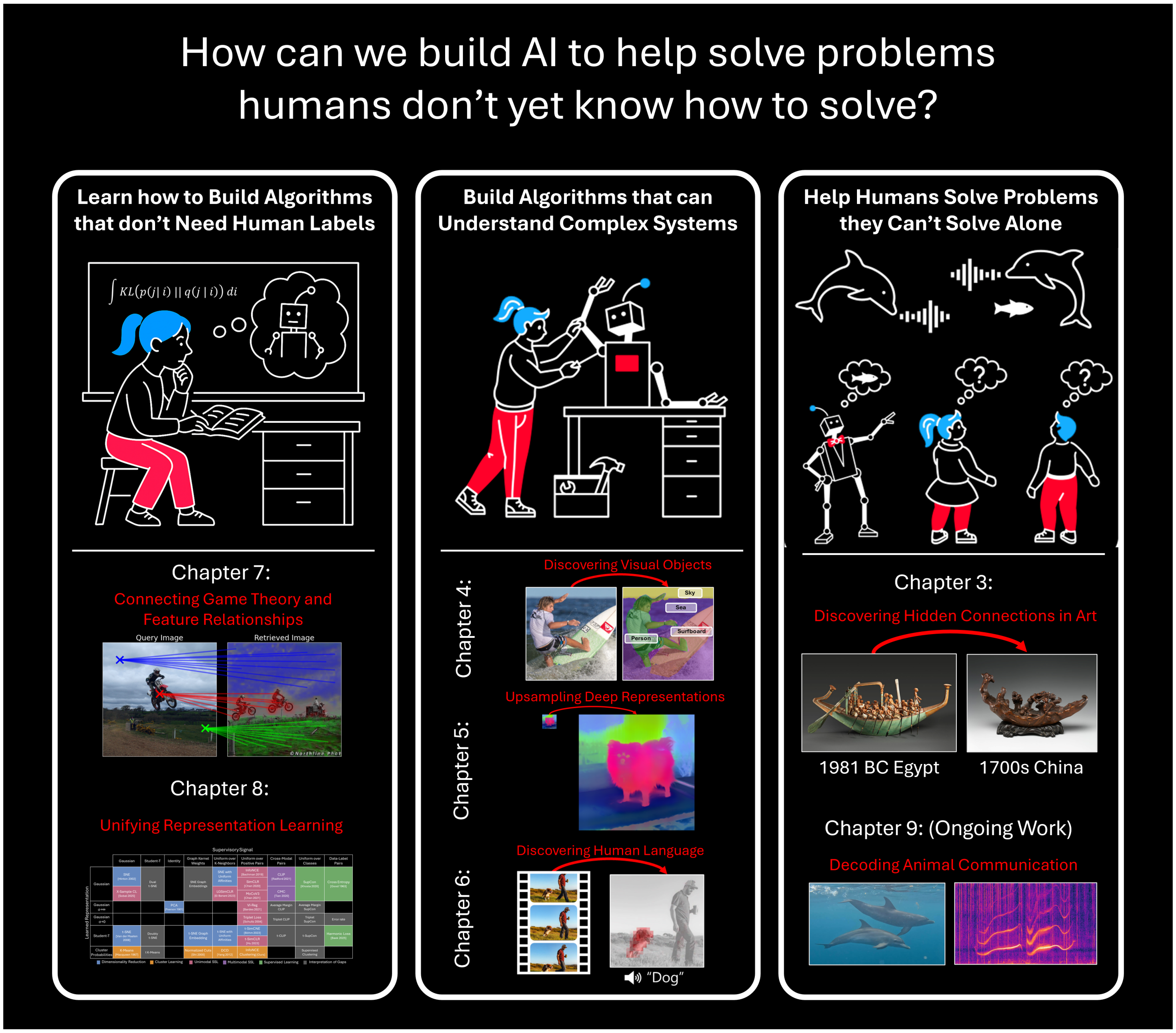

How can we build AI to help solve problems humans don't yet know how to solve?
Learn how to Build Algorithms that don't Need Human Labels
$\int KL(p(j|i)\,||\,q(j|i))\,di$
Chapter 7:
Connecting Game Theory and Feature Relationships
Query Image
Retrieved Image
Chapter 8:
Unifying Representation Learning
Build Algorithms that can Understand Complex Systems
Chapter 4:
Discovering Visual Objects
Sky
Sea
Person
Surfboard
Chapter 5:
Upsampling Deep Representations
Chapter 6:
Discovering Human Language
"Dog"
Help Humans Solve Problems they Can't Solve Alone
Chapter 3:
Discovering Hidden Connections in Art
1981 BC Egypt
1700s China
Chapter 9: (Ongoing Work)
Decoding Animal Communication

Figure 1.1: A graphical illustration of the content of this thesis. This work aims to understand how we can build algorithms that understand the world without human labels so we can help humans solve new problems. The thesis has three main thrusts, understanding the theory behind these algorithms, building these algorithms, and using them to discover interesting structure in the world.

1.3 Impact

This thesis expands what machines—and therefore humans—can understand. By introducing STEGO and DenseAV (Chapters 4 and 6) this thesis shows that fully unsupervised systems can discover objects and language without human supervision at any stage of the learning pipeline. These works dramatically improve on challenging benchmarks such as unsupervised and speech-prompted semantic segmentation by over 50%. These techniques have already been cited more than 400 times, starred by over 2.3 thousand developers on GitHub, and deployed in novel scientific domains such as medical imaging [1], LIDAR pointclouds [2], animal behavior [3], and nano-electronics [4]. By introducing FeatUp (Chapter 5) this thesis provides a key resource to the community to overcome the limited resolution of deep features in *any* vision backbone without altering the semantics of the network. FeatUp allows practitioners in any field to improve downstream task performance by up to 15% without retraining their systems (See Section 5.6). This thesis introduces a periodic table of representation learning (Chapter 8), which serves not just as a way to unify the field, but to guide the field towards a simple procedure to discover new classes of machine learning algorithms. I-Con has already been taught in lectures at MIT as a way to help students learn a broad class of algorithms within a single lecture.

In addition to impacting the fields of computer science and machine learning, this thesis has had broader corporate and societal impacts. Several chapters of this thesis (Chapters 3 and 7) formed the basis of the SynapseML project at Microsoft [5–8]. This project, which the author developed and led during this thesis, now has over 10,000 monthly active users, 125 contributors, and 8 million downloads. This project has also been awarded with a TIME Top 200 Invention of 2023 for its role in helping communities connect with literature, demonstrating impact for broader communities outside of computer science. The project enables industry-scale AI across a wide array of domains and embeds the fairness audits, first proposed in Chapter 7, in both customer-facing and internal Microsoft production systems [9]. Finally, Chapter 3 of this thesis was featured by the Metropolitan Museum of Art and Smithsonian Magazine for its role in helping museum visitors and art historians connect with different cultures and artistic media during the COVID-19 pandemic when physical museum spaces were closed. This application received tens of thousands of visitors from over 50 countries worldwide.

Finally, the thesis charts pathways for AI to collaborate with scientists on problems humans cannot solve alone. Chapter 9 describes ongoing collaborations to use the methods in Chapter 6 to create algorithms that are beginning to decode the communication of the Atlantic spotted dolphin from more than ten years of data collected by biologists and nonprofits. Similar ongoing work described in Chapter 9 work shows that large language models can match expert judgments in systematic reviews with 95% accuracy, saving researchers thousands of hours and laying the groundwork for algorithms that can automate science. This thesis lays the groundwork for AI systems that read, listen, and reason far beyond human limits while remaining accountable and accessible to the communities they serve.

1.4 Videos and Citations to Accompany Chapters

To make it easier to learn about the content of this thesis, Table 1.2 includes a series of short videos, web-based blog posts, and links to the original papers on which the chapters of this thesis are based. We also include a link to the PhD defense which can help new readers learn key material (especially chapters 3,5, and 7) within 45 minutes.

Chapters	**Resources and Publication**
4, 6, and 8	Thesis Presentation Video
3	*MosAIc: Finding Artistic Connections across Culture with Conditional Image Retrieval* Mark Hamilton, Stephanie Fu, Mindren Lu, Johnny Bui, Darius Bopp, Zhenbang Chen, Felix Tran, Margaret Wang, Marina Rogers, Lei Zhang, Chris Hoder, William T Freeman *NeurIPS Competition & Demonstration Track*, 2020 **Resources:** Paper, Video, Demo App
4	*Unsupervised Semantic Segmentation by Distilling Feature Correspondences* Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, William T Freeman *International Conference on Learning Representations (ICLR)*, 2022 **Resources:** Website with paper & video
5	*Featup: A Model-Agnostic Framework for Features at Any Resolution* Mark Hamilton, Stephanie Fu, Laura Brandt, Axel Feldman, Zhoutong Zhang, William T Freeman *International Conference on Learning Representations (ICLR)*, 2024 **Resources:** Website with paper & video
6	*Separating the "Chirp" from the "Chat": Self-Supervised Visual Grounding of Sound and Language* Mark Hamilton, Andrew Zisserman, John R Hershey, William T Freeman *Computer Vision and Pattern Recognition (CVPR)*, 2024 **Resources:** Website with paper & video
7	*Axiomatic Explanations for Visual Search, Retrieval, and Similarity Learning* Mark Hamilton, Scott Lundberg, Lei Zhang, Stephanie Fu, William T Freeman *International Conference on Learning Representations (ICLR)*, 2022 **Resources:** Website with paper & video
8	*I-Con: A Unifying Framework for Representation Learning* Shaden Naif Alshammari, John R Hershey, Axel Feldmann, William T Freeman, Mark Hamilton *International Conference on Learning Representations (ICLR)*, 2025 **Resources:** Website with paper & video

Table 1.2: Online resources and full publication details for each chapter.

Chapter 2

Preliminaries

This chapter provides a gentle introduction to the core ideas and mathematical objects that underpin the remainder of this thesis. The aim is to offer intuitive explanations and background on foundational concepts in machine learning, deep learning, and representation learning, especially for readers from different backgrounds. Readers already familiar with these areas may wish to skim or skip this section.

2.1 Vectors, Matrices, and Inner Products

Many of the ideas in machine learning and deep learning are most easily described using vectors and matrices. A vector is simply an ordered list of numbers, often written as $\mathbf{v} = [v_1, v_2, ..., v_n]$. Vectors are useful for representing data, such as the pixel values in an image or the "ideas" an algorithm comes up with about an object.

A matrix is a two-dimensional array of numbers, and can be thought of as a collection of vectors arranged in rows or columns. Matrices are fundamental tools for organizing data, and also for describing the weights inside machine learning models. Matrices can transform vectors through matrix-vector multiplication. This is a fundamental operation for machine learning algorithms that transform data.

The inner product (or dot product) between two vectors $\mathbf{a}$ and $\mathbf{b}$, written $\langle \mathbf{a}, \mathbf{b} \rangle$ or $\mathbf{a} \cdot \mathbf{b}$, is computed by multiplying corresponding elements and adding them up: $\sum_{i=1}^{n} a_i b_i$. Inner products play a key role in measuring similarity between vectors.

2.2 Probability

Probability is the mathematical study of uncertainty. It provides a way to quantify how likely an event is to occur, using numbers between 0 (impossible) and 1 (certain). In machine learning, probability helps us model and reason about data, randomness, and predictions.

2.3 Conditional Probability

Conditional probability describes the likelihood of an event given that we know something else has happened. It is written as $P(A \mid B)$, which means the probability of A given B. Conditional probability is fundamental for reasoning about dependencies between variables, and is widely used in probabilistic models and algorithms.

2.4 What is Machine Learning?

Machine learning is the study of algorithms that improve through experience. Rather than following explicit rules written by humans, machine learning algorithms "learn" patterns from data. These algorithms can be trained to perform tasks such as classifying images, translating languages, or recommending movies.

2.5 Supervised Learning

Supervised learning is a type of machine learning where the algorithm learns from examples that include both inputs and desired outputs. For instance, in image classification, the algorithm sees many images, each labeled with its correct category. The goal is to learn a rule that predicts the label for new, unseen images.

2.6 Self-Supervised and Unsupervised Learning

In unsupervised learning, the algorithm receives only raw data, without any labels or outputs. The aim is to discover structure or patterns in the data. For example, clustering algorithms try to group similar objects together.

Self-supervised learning is a recent and rapidly growing area where the learning algorithm generates its own supervision by solving auxiliary tasks. For instance, it might learn by predicting missing parts of an image, or by distinguishing whether two pieces of data are related.

2.7 Clustering

Clustering is an unsupervised learning technique used to group similar data points together based on their features. The goal is to discover structure in the data by organizing it into clusters, where items within a cluster are more similar to each other than to those in other clusters. Clustering methods are widely used in representation learning to analyze and evaluate the learned feature spaces.

2.8 Deep Learning

Deep learning refers to a set of machine learning techniques that algorithms with many layers (hence "deep") inspired by the human brain. These networks are composed of interconnected units, or "neurons", which can learn to represent highly complex patterns in data. Deep learning has enabled breakthroughs in computer vision, natural language processing, and many other fields.

2.9 Deep "Representations" or "Features"

A central idea in modern machine learning is that models learn to extract "features" or "representations" from raw data. In deep learning, each layer of a neural network transforms the data into a new representation. Early layers might detect simple patterns (like edges in images), while deeper layers capture more abstract concepts (like objects or actions). Representations usually take the form of vectors of data. These vectors are usually "learned" from the data and capture the "ideas" that a network has about a piece of data.

2.10 From Low Level Representations to Semantics

Initially, data is represented at a low level (e.g., pixel values in images, waveform samples in audio). As information passes through the layers of a deep network, the representations become increasingly abstract and semantic, eventually capturing high-level ideas such as the identity of an object, the sentiment of a sentence, or the meaning of a word.

2.11 Vectors as Carriers of Meaning or Semantics

In modern machine learning, vectors do not just store numbers—they can capture meaning or semantics about data. For example, the feature vector produced by a neural network for an image might encode the presence of objects, their shapes, or even more abstract concepts. In language models, word vectors can represent similarities in meaning: words with similar meanings often have feature vectors that are close together in the learned space. In this way, vectors act as mathematical stand-ins for the underlying ideas, objects, or meanings present in the data.

2.12 Inner Products and Cosine Similarity Between Features

One simple way to compare two feature vectors is to compute their inner product. A related measure is the cosine similarity, which takes the inner product of two vectors and divides by the product of their lengths (norms):

$$\text{cosine similarity}(\mathbf{a}, \mathbf{b}) = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

Cosine similarity ranges from -1 (opposite directions) to 1 (same direction), and is commonly used to measure how similar two representations are, regardless of their magnitudes.

This thesis will focus a great deal of effort into algorithms that compare features with inner products. Every chapter will cover an algorithm that compares features with inner products. This computation will effectively allow us to ask "how similar are these two objects or representations" on a scale that ranges from -1 (opposites) , 0 (unrelated), 1 (the same) and all values in between.

2.13 Contrastive Learning

Contrastive learning is a machine learning approach where models learn by comparing pairs of examples. The idea is to bring similar examples closer together in their learned representation, while pushing dissimilar examples further apart. For instance, different views of the same image are treated as similar, and images from different sources are treated as dissimilar.

This method is especially useful in self-supervised learning, where explicit labels are not available. By learning from comparisons, contrastive learning helps models discover useful and general representations from raw data.

2.14 Foundation Models and Backbones

Foundation models are large machine learning models trained on broad and diverse datasets, often at scale, to solve a wide range of tasks. These models, such as large language models and vision transformers, serve as general-purpose systems that can be adapted to many applications with little or no additional training.

The term *backbone* refers to the main part of a neural network used to extract features from data. In practice, backbones are often pre-trained models that provide useful representations for tasks like classification, retrieval, or segmentation. Many modern systems build on foundation models or use popular backbones as the starting point for more specialized models.

2.15 Transfer Learning

Transfer learning is a technique where a model trained on one task or dataset is reused as the starting point for a new task. Instead of training a new model from scratch, transfer learning leverages the knowledge already captured by an existing model, often leading to faster training and better performance, especially when labeled data is limited.

A common approach in deep learning is to use a pre-trained model as a feature extractor, and then train a simple classifier—called a *linear probe*—on top of these features for a specific task. The linear probe is typically just a single linear layer that learns to map the pre-trained representations to new labels. This strategy is widely used to evaluate how much useful information is captured by the learned representations of a model.

2.16 Neural Network Architectures

Neural network architectures define how the layers and connections in a network are organized. Two of the most important architectures in modern deep learning are convolutional neural networks (CNNs) and transformers.

Convolutional neural networks (CNNS) are especially well-suited for image and visual data. They use convolutional layers that scan small regions of the input, allowing them to detect patterns like edges, textures, and shapes. This makes CNNs very effective for image classification, object detection, and similar tasks.

Transformers, originally developed for natural language processing, have become popular across many domains, including vision and audio. Transformers rely on a mechanism called *self-attention*, which enables them to model relationships between all parts of the input, regardless of their position. This flexibility has made transformers the backbone of many recent advances in deep learning, such as large language models and powerful vision models.

The choice of architecture—CNN, transformer, or otherwise—can greatly affect a model's ability to learn effective representations for a given problem.

2.17 Image Retrieval and Search Engines

Image retrieval systems allow users to find images similar to a given query. Modern systems often work by comparing feature representations of images using inner products or cosine similarity. The general idea extends to search engines for text, audio, and other types of data: relevant results are those whose representations are most similar to the query's representation.

2.18 Multimodal Learning

Multimodal learning refers to building models that can process and combine information from different types of data, such as images, text, audio, or video. By learning joint representations that connect multiple modalities, these models can, for example, match captions to images or relate sounds to objects in a scene. Multimodal approaches are essential for tasks that require understanding and reasoning across different forms of information.

2.19 Model Explainability

As machine learning models become more complex, it becomes increasingly important to understand how they make decisions. Model explainability refers to a range of techniques designed to make the behavior of models more transparent to humans. For example, some methods highlight which parts of an image contributed most to a prediction, or explain why a particular item was retrieved by a search engine.

2.20 Information Theory

Information theory is a field of mathematics that studies how information is measured, transmitted, and compressed. It introduces concepts such as entropy, which quantifies the uncertainty or randomness in data, and mutual information, which measures how much knowing one variable tells us about another. Information theory provides the foundation for many ideas in machine learning, including how models learn, represent, and communicate information.

2.21 KL Divergence

The Kullback-Leibler (KL) divergence is a mathematical measure of how one probability distribution differs from another. If P and Q are two probability distributions, the KL divergence from Q to P is:

$$D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

KL divergence is widely used in machine learning to measure how well a model's predicted distribution matches the true distribution of data, and plays a central role in many algorithms for learning representations.

2.22 Game Theory

Game theory is a branch of mathematics that studies situations where multiple agents (or players) interact, each making decisions that affect the outcome for everyone involved. It provides tools to analyze how individuals or groups choose strategies, anticipate others' actions, and respond to incentives. Game theory has applications in economics, biology, computer science, and machine learning, especially when modeling competition or cooperation among agents.

2.23 Cooperative Game Theory

Cooperative game theory focuses on scenarios where agents can form groups, or coalitions, and work together to achieve shared goals. The central questions are how to assign credit or value to each participant and how to fairly distribute rewards among them. Key concepts from cooperative game theory, such as the Shapley value, have become important in model explainability—helping to determine how much each feature or input contributes to a machine learning model's prediction.

2.24 The Shapley Value

The Shapley value is a concept from cooperative game theory that provides a fair way to assign credit or importance to each participant in a group working together. In machine learning, the Shapley value is often used to measure how much each feature or input contributes to a model's prediction. It does this by considering all possible combinations of features and averaging their contributions.

A remarkable property of the Shapley value is its uniqueness: it is the only method that satisfies a specific set of fairness criteria—efficiency, symmetry, linearity, and the dummy property. These criteria capture the idea of fairness in several ways: the total credit is always fully distributed (efficiency), equal contributors receive equal credit (symmetry), irrelevant participants get zero credit (dummy property), and contributions add up consistently (linearity). Together, these properties ensure that the Shapley value gives a principled and fair assignment of credit to all participants or features.

2.25 The Harsanyi Dividend

The Harsanyi dividend generalizes the Shapley Value and measures the unique contribution made by a specific group of participants (or features) working together, beyond what any smaller subgroup can achieve on its own. In machine learning, Harsanyi dividends can help explain how combinations of features interact to influence a model's prediction, by isolating the effect that only arises when certain features are considered together. Shapley Values are in some sense a "first-order" Harsanyi Dividend.

Chapter 3

MosAIc: Relationships Between Visual Representations can find Connections in Art that Span Time and Culture

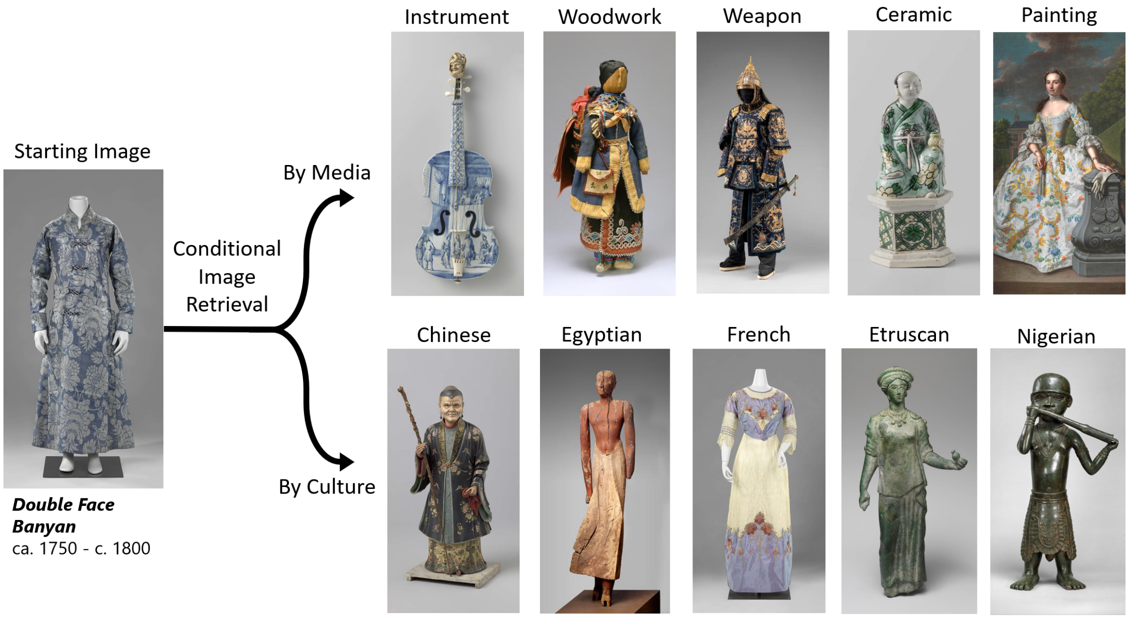


Figure 3.1: Conditional image retrieval results on artwork from the Metropolitan Museum of Art and Rijksmuseum using media and culture (text above images) as conditioners.

3.1 Website and Video

For a quick video overview and interactive demonstration of the material in this chapter, see our Video and Demo App.

3.2 Chapter Summary

This chapter marks a first exploration of possible applications that can be built by carefully analyzing the relationships between visual representations in large corpora of art. This chapter introduces “MosAIc”, an interactive web app that allows users to find pairs of semantically related artworks that span different cultures, media, and millennia. To create this application, we introduce Conditional Image Retrieval (CIR) which combines visual similarity search with user supplied filters or “conditions”. This technique allows one to find pairs of similar images that span distinct subsets of the image corpus. We provide a generic way to adapt existing image retrieval data-structures to this new domain and provide theoretical bounds on our approach’s efficiency. To quantify the performance of CIR systems, we introduce new datasets for evaluating CIR methods and show that CIR performs non-parametric style transfer. Finally, we demonstrate that our CIR data-structures can identify “blind spots” in Generative Adversarial Networks (GAN) where they fail to properly model the true data distribution.

3.3 Introduction

We begin this thesis with an analysis of deep image representations from both supervised and self-supervised networks. Although these types of algorithms are trained to solve simple tasks like classification or retrieval respectively, they learn deep representations of spectacular intelligence and fidelity. By asking the simple question: “What images have similar representations?” we can build apps like image search engines, recommendation systems, image de-duplication systems, and more. This chapter expands on these key ideas through the lens of charting the flow of ideas in the visual arts. We find that by modifying the core question to “Can I find a pair of images that arise in very different artistic settings (cultures, media, time periods, etc), yet have very similar representations?” we can build systems that discover uncanny, and historically interesting pairs of artworks that reflect broader flows of art over time. Most interestingly, this algorithm is totally unsupervised and requires no labels other than some basic metadata about where an artwork originated and what media the artwork uses. The system needs no knowledge of cultural exchange, human migration, or artistic provenance a-priori. This will be our first example of how hidden relationships between deep image features have the power to discover tangible relationships in large and complex datasets.

Image retrieval (IR) systems aim to find related images in a large corpora from any given query image. These systems power products like Google Image Search, Tin-Eye, product recommendations, and many other important applications. In many image retrieval applications, it is natural to limit the scope of the query to a subset of images. For example, returning similar clothes by a certain brand, or similar artwork from a specific artist. Currently, it is a challenge for IR systems to restrict their attention to sub-collections of images on the fly, especially if the subset is very distinct from the query image. This work explores how to create image retrieval systems that work in this setting, which we call “Conditional Image Retrieval" (CIR). We find that CIR can uncover pairs of artworks within the combined open-access collections of the Metropolitan Museum of Art [10] and the Rijksmuseum [11] that have striking visual and semantic similarities despite originating from vastly different cultures and millennia and introduce an interactive web app MosAIc (www.aka.ms/mosaic) to demonstrate the approach. To understand our methods better, we evaluate CIR on the FEI Face Database [12] as well as two new large-scale image datasets that we introduce to help evaluate these systems. These experiments show that CIR can perform a non-parametric variant of "style transfer" where neighbors in different subsets have similar content but are in the "style" of the target subset of images.

We also investigate ways to improve IR system performance in the conditional setting. One challenge current systems face is that a core component of many IR systems, K-Nearest Neighbor (KNN) data-structures, only support queries over the entire corpus. Restricting retrieved images to a particular class or filter requires filtering the "unconditional" query results, switching to brute force adaptively [13], or building a new KNN data-structure for each filter. The first approach is used in several production image search systems [14–16], but can be costly if the filter is specific, or the query image is far from valid images. Switching to brute force adaptively can mitigate this problem but is limited by the speed of brute force search, and its performance will degrade if the target subset far from the query point. Finally, maintaining a separate KNN data-structure for each potential subset of the data is costly and can result in 2^n data-structures, where n is the total number of images. In this work, we show that tree-based data-structures provide a natural way to improve the performance of CIR. More specifically, we prove that Random Projection Trees [17] can flexibly adapt to subsets of data through pruning. We use this insight to design a modification to existing tree-based KNN methods that allows them to quickly prune their structure to adapt to any subset of their original data using an inverted index. These structures outperform the commonly used CIR heuristics mentioned above. Finally, we investigate the structure of conditional KNN trees to show that they can reveal areas of poor convergence and diversity ("blind spots") in image based GANs. We summarize the contributions of this work as follows:

- We introduce an interactive web application to discover connections across cultures, artists, and media in the visual arts.
- We prove an efficiency lower bound for solving CIR with pruned Random Projection trees.
- We contribute a strategy for extending existing KNN data-structures to allow users to efficiently filter resulting neighbors using arbitrary logical predicates, enabling efficient CIR.
- We show that CIR data-structures can discover "blind spots" where GANs fail to match the true data.

3.4 Background

IR systems aim to retrieve a list of relevant images that are related to a query image. "Relevance" in IR systems often refers to the "semantics" of the image such as its content, objects, or meaning. Many existing IR systems map images to "feature space" where distance better corresponds to relevance. In feature space, KNN can provide a ranked list of relevant

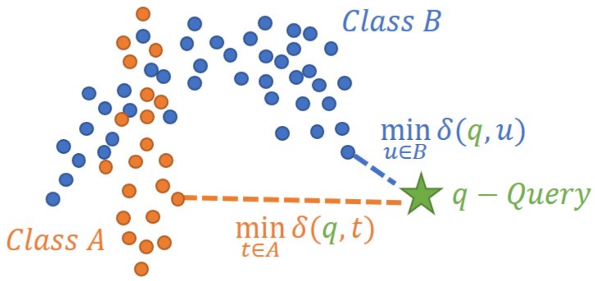


Figure 3.2: Conditional K-Nearest Neighbors for a query point, q, and distance, δ, on a simple two class dataset.

Component	Space Efficiency	Measured
Data	$\mathcal{O}(n \times d)$	16 GB
Tree	$\mathcal{O}((2n/l) \times d)$	65 MB
Cond. Index	$\mathcal{O}(c \times 2n/l)$	6.4 MB

Table 3.1: Space efficiency of a binary CKNN Tree with number of points, n, dimensionality, d, leaf size l, and number of classes in the index, c. Measured results are from a tree built on the Conditional Art dataset: $n = 1000000, d = 2048, l = 500, c = 200$.

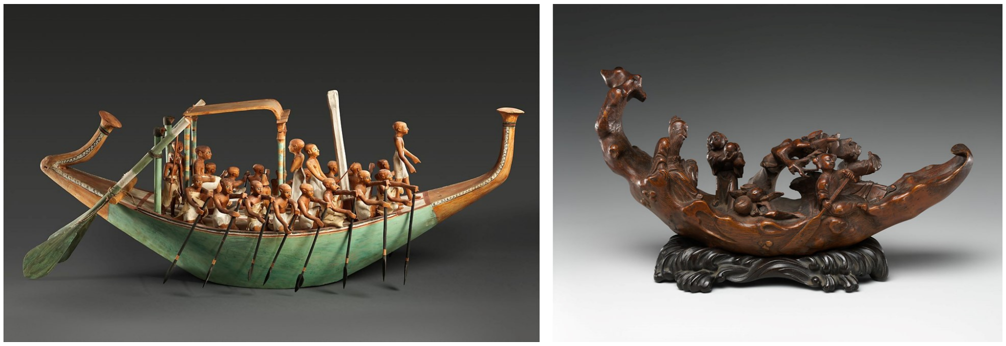


Figure 3.3: A pair of cross cultural images found with CIR. Left: *Model Paddling Boat* from 1980 BC Egypt. Right: *Immortal Raft* from 18th Century China.

images [18]. Good features and distance metrics aim to align with our intuitive senses of similarity between data [19] and show invariance to certain forms of noise [20]. There is a considerable body of work on learning good "features" for images [21–25]. In this work we leverage features from intermediate layers of deep supervised models, which perform well in a variety of contexts and are ubiquitous throughout the literature. Nevertheless, our methods could apply to any features found in the literature including those from collaborative filtering, text, sound, and tabular data.

There are a wide variety of KNN algorithms, each with their own strengths and weaknesses. Typically, these methods are either tree-based, graph-based, or hash-based [26]. Tree-based methods partition target points into hierarchical subsets based on their spatial geometry and include techniques such as the KD Tree [27], PCA Tree [28], Ball Tree [29], some inverted index approaches [30], and tree ensemble approaches [31]. Some tree-based data-structures allow exact search with formal guarantees on their performance [17]. Graph-based methods rely on greedily traversing an approximate KNN graph of the data, and have gained popularity due to their superior performance in the approximate NN domain [26, 32]. There are many hash-based approaches in the literature and [33] provides a systematic overview. To our knowledge, neither graph nor hash-based retrieval methods can guarantee finding the nearest neighbor deterministically. However, fast approximate search is often sufficient for many applications. In our work we focus on tree-based methods because it is unclear how to create an analogous method for graph-based data-structures. Nevertheless, tree-based methods are

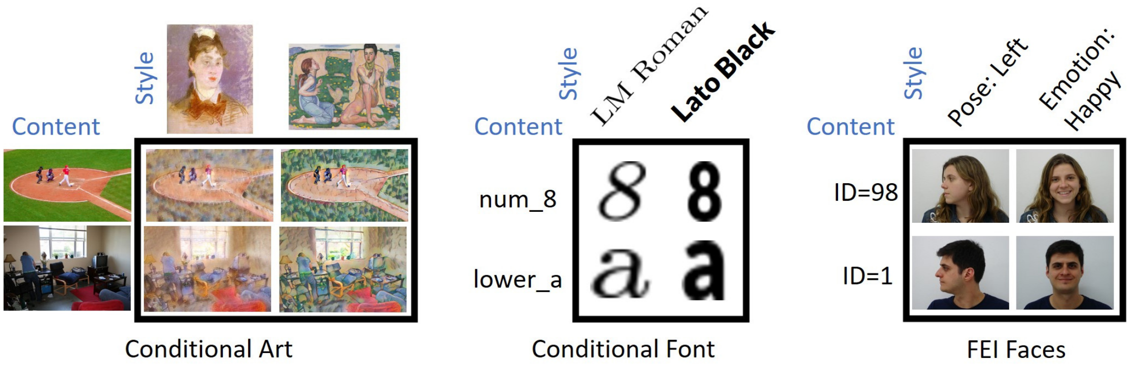


Figure 3.4: Representative samples from the ConditionalArt dataset (left), ConditionalFont dataset (middle), and FEI Face dataset (right). CIR systems conditioned on style should retrieve images of the same content.

widely used, especially when exact results are needed. Surprisingly, conditional KNN systems have only received attention recently, even though conditional queries appear in shopping, search, and recommendation systems. To our knowledge, [13] is the only effort to improve performance of these systems by adaptively switching from a “query-then-filter" strategy to brute-force at a particular size threshold.

3.5 Conditional Image Retrieval

To generalize an IR system to handle queries over any image subset we generalize the KNN problem to this setting. More formally, the Conditional K-Nearest Neighbors (CKNNs) of a query point, q, are the k closest points with respect to the distance function, δ, that satisfy a given logical predicate (condition), $\mathcal{S}$. We represent this condition as a subset of the full corpus of points, $\mathcal{X}$:

$$CNN(q, \mathcal{S} \subseteq \mathcal{X}) = \operatorname*{argmin}_{t \in \mathcal{S}} \delta(q, t)$$

When the conditioner, $\mathcal{S}$, equals the full space, $\mathcal{X}$, we recover the standard KNN definition. Figure 3.2 shows a visualization of CKNN for a two-dimensional dataset with two classes. With conditional KNN queries it's possible to combine logical predicates and filters with geometry-based ranking and retrieval.

To create a CIR system, one can map images to a “feature-space”, where distance is semantically meaningful, prior to finding CKNNs. One of the most common featurization strategies uses the penultimate activations of a supervised network such as ResNet50 [34] trained on ImageNet [35]. Alternatively, using “style” based features from methods like AdaIN [36] enable CIR systems that retrieve images by “style” as opposed to content. Deep features capture many aspects of image semantics such as texture, color, content, and pose [37] and KNNs in deep feature space are often both visually and semantically related. We aim to explore whether this observation holds for conditional matches across disparate subsets of images, which requires a more global feature-space consistency.

		Featurization Method								
Dataset	Metric	RN50	RN101	MN	SN	DN	RNext	dlv3101	MRCNN	Random
CA	@1	.50	.51	.55	.44	**.59**	.46	.37	.45	.0002
	@10	.70	.68	.71	.62	**.76**	.65	.55	.63	.002
CF	@1	.41	.37	.39	**.44**	.43	.38	.33	**.44**	.016
	@10	.77	.76	.76	**.80**	.79	.76	.73	.79	.16
FEI	@1	.80	.84	.85	.79	**.87**	.86	.72	.78	.005
	@10	.94	.93	.94	.89	**.95**	.94	.86	.92	.05

Table 3.2: Performance of CIR (Accuracy @N) on content recovery across style variations for both the ConditionalFont (CF) and ConditionalArt (CA) datasets using a variety of features from pre-trained networks. Results show CIR retrieves the same content image across different styles. For full details on experimental conditions see Section 3.13

.

3.6 Discovering Shared Structure in Visual Art

We find that CIR on the combined Met and Rijksmusem collections finds striking connections between art from different histories and mediums. These matches show that even across large gaps in culture and time CIR systems can find relevant visual and semantic relations between images. For example, Figure 3.3 demonstrates a pair of images that, despite being separated by 3 millennia and 7,000 Kilometers, have an uncanny visual similarity and cultural meaning. More specifically, both works play a role in celebrating and safeguarding passage into the afterlife [38–40]. Matches between cultures also highlight cultural exchange and shared inspiration. For example, the similar ornamentation of the Dutch Double Face Banyan (left) and the Chinese ceramic figurine (top row second from left) of Figure 3.1 can be traced to the flow of porcelain and iconography from Chinese to Dutch markets during the 16th-20th centuries [41, 42]. CIR also provides a means for diversifying the results of visual search engines through highlighting conditional matches for cultures, media, or artists that are less frequently explored. We hope CIR can help the art-historical community and the public explore new artistic traditions. This is especially important during the COVID-19 pandemic as many cultural institutions cannot accept visitors. To this end, we introduce an interactive art CIR application, aka.ms/mosaic, and provide more details in Section 3.7. In Section A.2 of the Appendix we also provide additional examples and representative samples.

3.7 The MosAIc Web Application

As an application of CIR for the public, we introduce MosAIc (aka.ms/mosaic), a website that allows users to explore art matches conditioned on culture and medium. Our website aims to show how conditional image retrieval can find surprising and uncanny pairs of artworks that span millennia. We also aim to make it easy for interested users to find new artworks in cultures they might not think to explore during a physical museum visit. Using the MosAIc application, users can choose from a wide array of example objects to use as conditional search queries as shown in the left panel of Figure 3.5. Users can select from

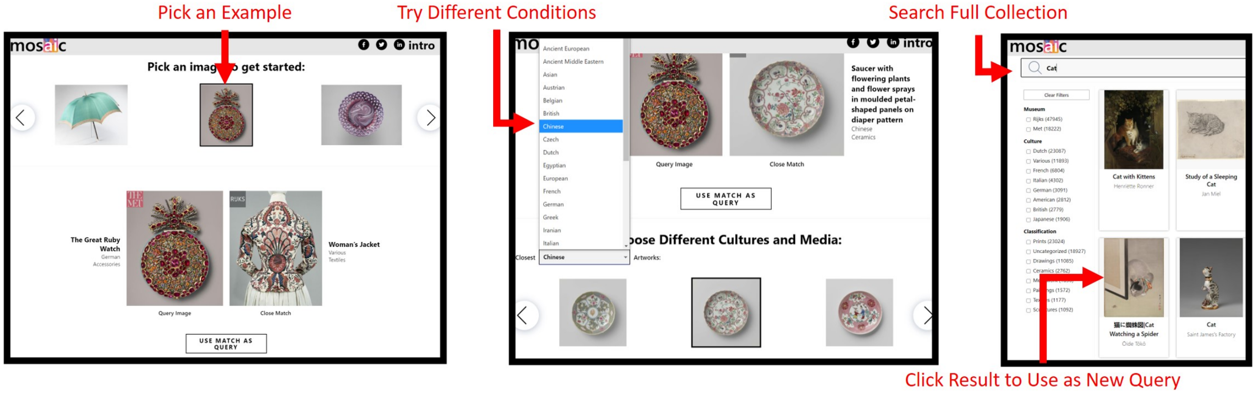


Figure 3.5: Using the MosAIc web application (aka.ms/mosaic). After watching a short video explaining the app, users can select a work of art to find conditional matches with (left). Users can find conditional matches for a variety of different cultures and media (middle). To further explore the collection, users can search for new query objects using a conventional search index (right). Users can also construct chains of conditional matches using the "Use match as query" button below the main matches.

an array of different cultures and media to condition their searches as in Figure 3.5 middle. Selecting a specific medium or culture, allows the user to browse the top conditional matches in that category and use these matches as new query images. This enables traversing the collection using conditional searches to find relevant content in different areas of the collection. Additionally, for users who want to use a specific work of art as a starting point we have added a conventional text based search engine to quickly find specific works relating to a keyword as in Figure 3.5 right.

The mosaic application combines a React [43] front-end with a back-end built from Azure Kubernetes Service, Azure Search, and Azure App Services. Our front-end features responsive design principles to support for mobile, tablet, desktop, and ultra-wide displays. We also aim to use high-contrast design to make the application more accessible to the low-vision community. To create the conditional search index, we featurize the combines Metropolitan Museum of Art and Rijksmuseum open access collections using ResNet50 from torchvision [44]. We then add these features to a Conditional Ball tree for real-time conditional retrieval and deploy this method as a RESTful service on Azure Kubernetes Service. Additionally, we store image metadata, automatically generated image captions, and detected objects in an Azure Search index which allows querying for additional information, and supports text search. To add captions and detected objects to over 500k images we use the Cognitive Services for Big Data [6].

3.8 Evaluating CIR Quality

Though finding connections between art is of great importance to the curatorial and historical communities, it is difficult to measure a system's success on this dataset as there are no ground truth on which images *should* match. To understand the behavior of CIR systems quantitatively we investigate datasets with known content images aligned across several different "styles" or subsets to retrieve across. More specifically, if the conditioning information represents the image "style" and the features represent the "content", CIR should find an image with the same content, but constrained to the style of the conditioner, such as "Ceramic" or "Egyptian" in Figure 3.1. Through this lens, CIR systems can act as "non-parametric" style transfer systems. This approach differs from existing style transfer and visual analogy methods in the literature [36, 45] as it does not generate new images, but rather it finds analogous images within an existing corpora.

To this end, we apply CIR to the FEI face database of 2800 high resolution faces across 200 participants and 14 poses, emotions, and lighting conditions. We also introduce two new datasets with known style and content annotations: the ConditionalFont and ConditionalArt datasets. The ConditionalFont dataset contains 15687 32×32 grayscale images of 63 ASCII characters (content) across 249 fonts (style). The ConditionalArt dataset contains 1,000,000 color images of varying resolution formed by stylizing 5000 content images from the MS COCO [46] dataset with 200 style images from the WikiArt dataset [47] using an Adaptive Instance Normalization [36]. Although this dataset is "synthetic", [48] show that neural style transfer methods align with human intuition. We show representative samples from each dataset in Figure 3.4.

With these datasets it's possible to measure how CIR features, metrics, and query strategies affect CIR's ability to match content across styles. To measure retrieval accuracy, we sampled 10000 random query images. For each random query image, we use CIR to retrieve the query image's KNNs conditioned on a random style. We then check whether any retrieved images have the same content as the original query image. In Table 3.2, we explore how the choice of featurization algorithm affects CIR systems. All methods outperform the random baseline of Table 3.2, indicating that they are implicitly performing non-parametric content-style transfer. DenseNet (DN) [49] and Squeezenet (SN) [50] tend to perform well across all datasets. CIR performs well across all three tasks *without* fine tuning to the structure of the datasets, indicating that this approach can apply to other zero-shot image-to-image matching problems.

3.9 Fast CKNN with Adaptive Tree Pruning

In Section 3.8 we have shown that CIR is semantically meaningful in several different contexts, but the question remains as to whether this approach affords an efficient implementation that can scale to large datasets with low latency. Conventional IR systems scale to this setting using dedicated data-structures such as trees, spatial hashes, or graphs. There are a wide variety of strategies with provable guarantees in the unconditional setting, but it is not known if existing data-structures can apply naturally to the conditional setting. In this work we focus on extending tree-based methods to the conditional setting. Tree-based methods are some of the only methods that guarantee *exact* KNN retrieval, and there are already several theoretical results on the performance of these methods [17, 51]. In particular, [17] show that RandomProjection-Max (RP) trees can adapt to the intrinsic dimensionality of the data and prove bounds that demonstrate the effectiveness of the data-structure. [51] continue this line of reasoning and prove a packing lemma using a bound on the aspect ratio of RP tree

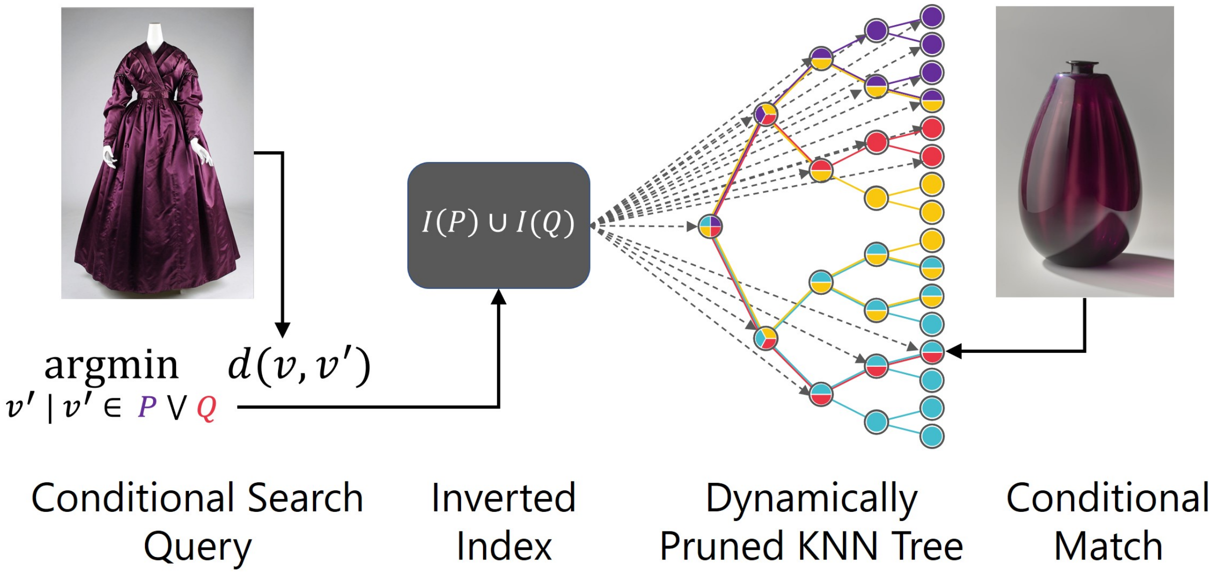


Figure 3.6: Dynamic tree pruning based CIR architecture. The user specified condition, $P \vee Q$, is translated to an inverted index query and the result is used to prune the unconditional KNN tree where nodes are colored based on which conditions they contain. This pruned tree accelerates conditional search for any subset by reducing the number of nodes considered in tree traversal.

cells. These works show that RP trees are effective at capturing the geometry of the training data. Our aim is to show that they also capture the geometry of *subsets* of the training data through their sub-trees. More specifically, we show that for any subset of the training data, one can derive probabilistic bound the number of nodes in the tree that contain this subset. More formally:

Theorem 3.9.1. *Suppose an RPTree-Max, $\mathcal{T}$, is built using a dataset $\mathcal{X} \subset \mathbb{R}^D$, of diameter W, with doubling dimension $\leq d$. Further suppose $\mathcal{T}$ is balanced with a cell-size reduction rate bounded above by γ. Let $\mathcal{S} \subseteq \mathcal{X}$ be a subset of the dataset used to build the tree and $\mathcal{B}$ a finite set of radius $R > 0$ balls that cover $\mathcal{S}$. For every $0 < \epsilon < 1$ there exists a constant, $c > 0$, such that with probability $> 1 - \epsilon$ the fraction of cells that contain points within $\mathcal{S}$ is bounded above by $|\mathcal{B}|2^{-log_\gamma(W/R')}$ where $R' = cRd\sqrt{d}\log(d)$*

We point readers to [51], for the precise definition of an RPTree, cell-size, and the doubling dimension. To sketch the proof, we first generalize an aspect bound from [51] to show that, with high probability, small radius balls can be completely inscribed within small radius RP tree cells. Because it takes several levels before the tree's cells shrink to this size, we can bound this cell's depth and thus the size of its sub-tree relative to the full tree. By considering a collection of balls that cover our target subset, we arrive at the final bound. See section A.3 of the Appendix for a full proof.

This theorem not only shows that sub-trees of an RP tree capture the geometry of training dataset subsets, but also points to a method to improve the speed of CKNN. Namely, we can prune tree nodes that do not hold points within our target subset prior searching for

```
input  : A point, $q$, a condition, $\mathcal{S} \subseteq \mathcal{X}$, a tree, $root$, and an inverted index, $I$
output : Closest point, $p^* \in \mathcal{S}$, to $q$
$validNodes \leftarrow \bigcup_{s \in \mathcal{S}} I(s)$; $p^* \leftarrow$ null
def SearchNode($n$):
    if $n \in validNodes$ then
        if $n$ is a leaf node then
            $p \leftarrow$ closest point in $\mathcal{S}$
            if $d(p,q) < d(p^*,q)$ then
                $p^* \leftarrow p$
            end
        else
            $potentials \leftarrow$ children of $n$ which could hold a closer point
            for $child$ in $potentials$ do
                SearchNode($child$)
            end
        end
    end
SearchNode($root$); return $p*$
```

Algorithm 1: Querying a CKNN Tree

conditional neighbors. We diagram this procedure in Figure 3.6, and provide pseudo-code in Algorithm 1. We now turn our attention to quickly computing the proper sub-trees for each subset of the data. To this end, one can use an inverted index [52], I, that maps points, $x \in \mathcal{X}$ to the collection of their dominating nodes, $I(x) = \{n : x \text{ below node } n\}$. One can compute the subset of nodes that remain after pruning by taking the union of dominating nodes as shown in the first line of Algorithm 1 and in the illustration of the full search architecture in Figure 3.6. Evaluating the predicate on points within leaf nodes can also reduce computation.

Additionally, if the predicates of interest have additional structure, such as representing class labels, one can define a smaller class-based inverted index, $I_{class}(c)$ which maps a class label, c, to the set of dominating nodes. For these predicates, union and intersection operators commute through the class-based inverted index:

$$\begin{aligned} I(\mathcal{S}_a \cap \mathcal{S}_b) &= I_{class}(a) \cap I_{class}(b) \\ I(\mathcal{S}_a \cup \mathcal{S}_b) &= I_{class}(a) \cup I_{class}(b) \end{aligned} \tag{3.1}$$

where $\mathcal{S}_a$ is the subset of points with label a. This principle speeds a broad class of queries and accelerates document retrieval frameworks like ElasticSearch [53] and its backbone, Lucene [54]. We stress that this approach does not use Lucene to filter images or documents directly, but rather to filter nodes of a KNN retrieval data-structure at query time. This enables a rich “predicate push-down“ [55, 56] logic for KNN methods independent of how the tree splits points (Ball, Hyperplane, Cluster), the branching factor, and the topology of the tree. It also applies to ensembles of trees and to multi-probe LSH methods by pruning hash buckets. We note that our proposed indexing structure is small compared to the size

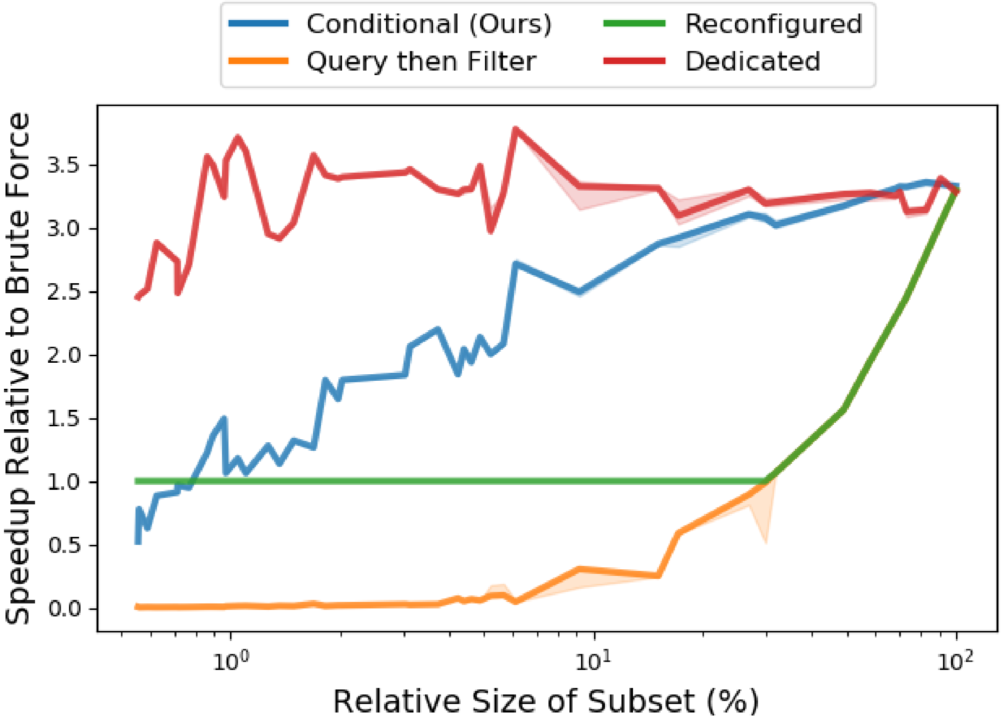


Figure 3.7: Query time of Conditional KNN approaches. Our approach (Conditional) achieves query performance approaching that of an tree recreated specifically for each query (Dedicated) *without the expensive re-creation cost*, and does not perform poorly with small conditions like "Query then Filter" strategies. Furthermore, our method accelerates queries across much smaller subsets than the reconfiguration strategy of [13]. Please see Section 3.13 and 3.9 for method details.

of the underlying dataset, and unconditional KNN tree, and provide an analysis of memory footprints in Table 3.1.

3.10 Performance

In Figure 3.7, we show the relative performance of several strategies for CIR on 488k Resnet50-featurized images ($dim = 2048$) from the combined MET and Rijksmusem open-access collections with a randomly chosen test set ($n = 1000$). We condition on artwork media, culture, and several combinations of these to create a variety of condition sizes. We measure the speedup compared to a vectorized Brute-Force search using NumPy arrays [57]. We implement CKNN methods with respect to one of the most used implementations of KNN, Sci-kit Learn's Ball Tree algorithm [58]. We compare our approach (Conditional) to, the standard "query-then-filter" approach, and adaptive switching to brute force search (Reconfigured) [13]. Finally, we compare to a "best-case" scenario of a KNN data-structure pre-computed for every subset (Dedicated). Though in practice it is often impossible to make an index for each subset, this setting provides an upper bound on the performance of any approach. Our analysis shows that adaptive pruning (Conditional) outperforms

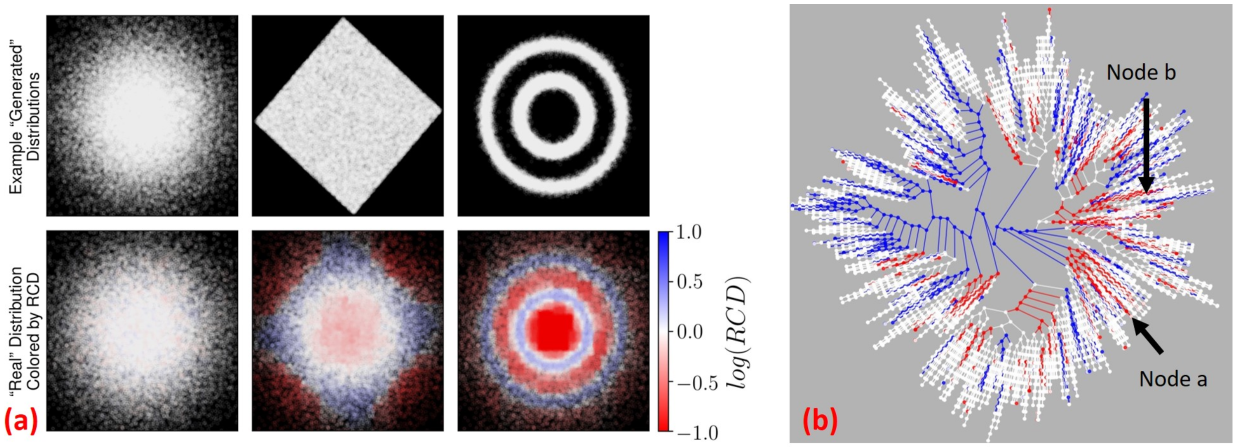


Figure 3.8: (a): Visualization of the RCD between several example distributions and a standard normal of "real" data ($n = 50k$). Upper plots show generated distributions, and bottom plots show the "real" distribution colored by the RCD induced by a CKNN Tree. Even though these datasets are identical under the popular Frechét inception distance (FID), the RCD detects areas where generated data over (blue) and under (red) samples the real data. (b): Nodes of a CKNN tree (Center node is the root) colored by statistically significant deviations of RCD from 1 ($p < 0.01$). This shows widespread differences between GAN outputs and true data. Red nodes represent areas where the GAN under samples the empirical distribution, and blue nodes over-sample. High discrepancy nodes a and b from Figure 3.9 are annotated.

other approaches and is close to optimal for large subsets of the dataset. Additionally, the performance of the "Query-then-filter" strategy quickly degrades for small subsets of the dataset as expected. Our approach is also compatible with prior work on adaptively switching to brute force and allows one to set the "switch-point" over 10x lower. We also note that these results hold with randomized conditions, and across other similar datasets.

Finally, we stress that the goal of this work is **not** to make the fastest unsupervised KNN method, but rather to evaluate generic strategies to transform these approaches to the conditional setting. There is a considerable body of work on fast, *approximate, unconditional* KNN methods which often outperform Scikit Learn's exact retrieval algorithms. We point readers to [26] for more details. We stress that exhaustive benchmarking of unconditional KNN indices and approaches is outside the scope of this work. For implementation, experimentation, environment, and computing details please see Section 3.13.

3.10.1 Implementation

We implement adaptive tree pruning for the existing Ball Tree and KD tree implementations in the popular SciKit-learn framework. Our implementation supports exact retrieval with several metrics, OpenMP parallelization [59], and Cython acceleration[60]. We also provide accelerations such as dense bit-array set operations, and caching node subsets on repeated conditioner queries. For larger scale datasets, we contribute a Spark based implementation of a Conditional Ball Tree to Microsoft ML for Apache Spark [7, 61].

To enable integration with differentiable architectures common in the community, we provide a high-throughput, PyTorch module [62] for CIR. This implementation is fundamentally brute force but uses Einstein-summation to retrieve conditional neighbors for multiple queries and multiple conditions simultaneously, and can increase throughput by over 100× compared to naive PyTorch implementations.

3.11 Limitations

This work does not aim to create the fastest KNN algorithm, but rather presents a formally motivated technique to speed up existing tree-based KNN methods in the conditional setting. KNN retrieval chooses some items significantly more than others, due to effects such as the "hubness problem" and we direct readers to [63] for possible solutions. We present additional diversity reducing geometries in Section A.1 of the Appendix. Our approach does not modify the KNN construction, simply prunes it afterwards. This may not be the most efficient solution when conditioner sizes are small, but it is orders of magnitude faster than recreating the tree. We also note that the performance of our conditional KNN methods are dependent on the underlying unconditional KNN tree, which often performs better on datasets with smaller intrinsic dimension.

3.12 Discovering "Blind Spots" in GANs

Efficient high-dimensional KNN search data-structures adapt to the geometry and intrinsic dimensionality of the dataset [17, 51]. Moreover, some recent KNN methods use approaches from unsupervised learning like hierarchical clustering [64] and slicing along PCA directions [28]. In this light, CKNN trees allow us to measure and visualize the "heterogeneity" of conditioning information within a larger dataset. More specifically, by analyzing the relative frequency of labels within the nodes of a CKNN tree, one can find areas with abnormally high and low label density. More formally, we introduce the Relative Conditioner Density (RCD) to measure the degree of over or under representation of a class c with corresponding subset $\mathcal{S}_c \subseteq \mathcal{X}$, at node n in the KNN tree:

$$RCD(n, c) = \frac{|n \cap \mathcal{S}_c|}{|n|} \frac{|\mathcal{X}|}{|\mathcal{S}_c|} \tag{3.2}$$

Here, $|n|$ is the number of points below node n in the tree. The RCD measures how much a node's empirical distribution of labels differs from that of the full dataset. $RCD > 1$ occurs when the node over-represents class c, and $RCD < 1$ occurs when the node under-represents a class, c. We apply this statistic to understand how samples from generative models, such as image-based GANs, differ from true data. In particular, one can form a conditional tree containing true data and generated samples, each with their own classes, c_t and c_g respectively. In this context, nodes with $RCD(\cdot, c_g) \ll 1$ are regions of space where the network under-represents the real dataset. To illustrate this effect, Figure 3.8a shows several simple 2d examples. Even though these datasets are identical with respect to the Fréchet Distance [65], coloring points based on their parent node RCD's can highlight areas

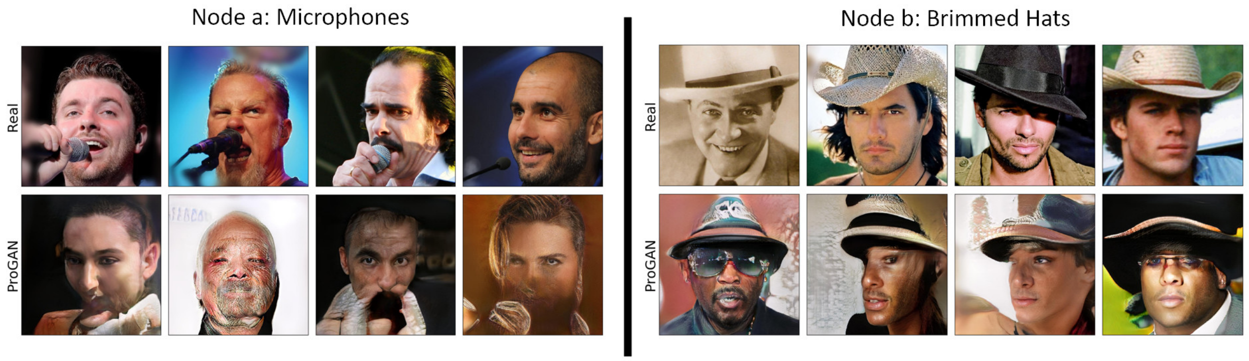


Figure 3.9: Samples from two statistically significant nodes from Figure 3.8. Images are randomly chosen and representative of those found at the node. Almost every real image in Node a contains microphones whereas no GAN generated outputs could create a microphone. Node b shows a clear bias towards brimmed hats, and the GAN samples have significant visual artifacts.

of over and under sampling of the true distribution by each "generated" distribution. In Figure 3.8b, we form a CKNN tree on samples from a trained Progressive GAN [66] and it's training dataset, CelebA HQ [67]. Coloring the nodes by RCD reveals a considerable amount of statistically significant structural differences between the two distributions. By simply thresholding the RCD (< 0.6), we find types of images that GANs struggle to reproduce. We show samples from two low-RCD nodes in Figure 3.9 and also note their location in Figure 3.8b. Within these nodes, Progressive GAN struggles to generate realistic images of brimmed hats and microphones. Though we do not focus this work on thoroughly investigating issues of diversity in GANs, this suggests GANs have difficulty representing data that is not in the majority. This aligns with the findings of [68], without requiring GAN inversion, additional object detection labels, or a semantic segmentation ontology. Furthermore, we note that the FID cannot capture the full richness of why two distributions differ, as this metric just measures differences between high dimensional means and co-variances. Using CKNN trees can offer more flexible and interpretable ways to understand the differences between two high dimensional distributions.

3.13 Experimental Details

All experiments use an Ubuntu 16.04 Azure NV24 Virtual Machine with Python 3.7 and scikit-learn v0.22.2 [58]. We use scikit-learn's Ball Tree and KD Tree and use numpy v1.18.1 [57] for brute force retrieval. For query-then-filter strategies we first retrieve 50 points, then increase geometrically (x5) if the query yeilds no valid matches. To form image features for Table 3.2, we use trained networks from torchvision v0.6 [44]. In particular, we use ResNet50 (RN50) [34], ResNet101 (RN101), MobileNetV2 (MN) [69], SqueezeNet (SN) [50], DenseNet (DN) [49], ResNeXt (RNext) [70], DeepLabV3 ResNet101 (dlv3101) [71], and Mask R-CNN (MRCNN) [72]. Features are taken from the penultimate layer of the backbone, and the matches of Table 3.2 are computed with respect to cosine distance. We use trained a

Progressive GAN from the open-source Tensorflow implementation accompanying [66].

3.14 Related Work

Image retrieval and nearest neighbor methods have been thoroughly studied in the literature, but we note that the conditional setting has only received attention recently. There are several survey works on KNN retrieval, but they only mention unconditional varieties [33, 73]. [74] has studied the mathematical properties of conditional nearest neighbor classifiers but works primarily with graph based methods as opposed to trees. They do not apply this to modern deep features and do not aim to improve query speed. There are a wide variety featurization strategies for IR systems. Gordo et. al [20] learn features optimized for IR. Siamese networks such as FaceNet embed data using tuples of two data and a similarity score and preserving this similarity in the embedding [24, 75]. Features from these methods could improve CIR systems. Conditional Similarity Networks augment tuple embedding approaches with the ability to handle different notions of similarity with different embedding dimensions [76]. This models conditions as similarities but does not generically restrict the search space of retrieved images to match a user's query. These features have potential to yield neighbor trees that, when pruned, have a similar structure and performance to dedicated trees. Sketch-based IR uses line-drawings as query-images but does not aim to restrict the set of candidate images generically [77]. Style transfer [78] and visual analogies [79] yield results like our art exploration tool but generate the analogous images rather than retrieve them from an existing corpus. [80] split IR systems into conditional subsystems, but do not tackle generic conditioners or provide experimental evaluation. [81] create an IR system conditioned on text input, but do not address the problem of generically filtering results. [82] and [83] respectively learn and use a hierarchy of concepts concurrently with IR features, which could be a compelling way to *learn* useful conditions for a Conditional IR system.

3.15 Chapter Conclusion

We have shown that Conditional Image Retrieval yields new ways to find visually and semantically similar images across corpora. We presented a novel approach for discovering hidden connections in large corpora of art and have creates an interactive web application, MosAIc to allow the public to explore the technique. We have shown that CIR performs non-parametric style transfer on the FEI faces and two newly introduced datasets. We proved a bound on the number of nodes that can be pruned from RandomProjection trees when focusing on subsets of the training data and used this insight to develop a general strategy for generalizing tree-based KNN methods to the conditional setting. We demonstrated that this approach speeds conditional queries and outperforms baselines. Lastly, we showed that CKNN data-structures can find and quantify subtle discrepancies between high dimensional distributions and used this approach to identify several "blind spots" in the ProGAN network trained on CelebA HQ.

Chapter 4

STEGO: Fully Unsupervised Semantic Segmentation by Distilling Relationships between Visual Representations

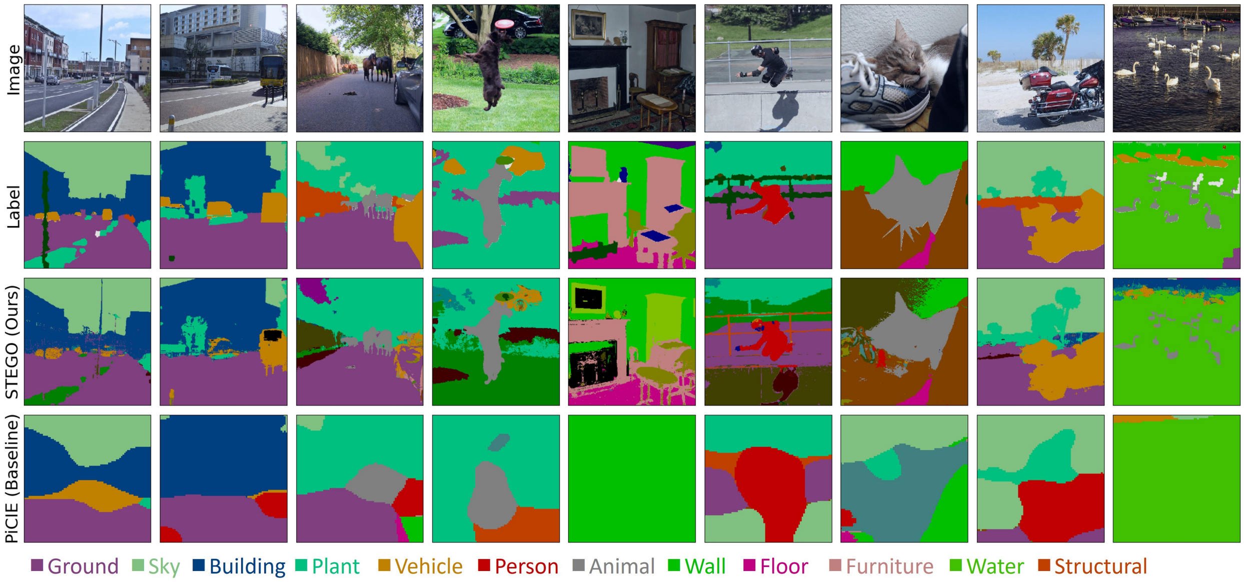


Figure 4.1: Unsupervised semantic segmentation predictions on the CocoStuff [84] 27 class segmentation challenge. Our method, STEGO, does not use labels to discover and segment consistent objects. Unlike the prior state of the art, PiCIE [85], STEGO's predictions are consistent, detailed, and do not omit key objects.

4.1 Website and Video

For a quick video overview and blog post of this chapter, see https://mhamilton.net/stego.html

4.2 Chapter Summary

In this chapter we show that relationships between features encode much more than just similarity between very different images (Chapter 3). Amazingly, this chapter shows that relationships between features can be used to rediscover visual objects and classify every pixel of the world without any human guidance or labels. We focus on the task of Unsupervised semantic segmentation, which aims to discover and localize semantically meaningful categories within image corpora without any form of annotation. To solve this task, algorithms must produce features for every pixel that are both semantically meaningful and compact enough to form distinct clusters. Unlike previous works which achieve this with a single end-to-end framework, we propose to separate feature learning from cluster compactification. Empirically, we show that current unsupervised feature learning frameworks already generate dense features whose correlations are semantically consistent. This observation motivates us to design STEGO (**S**elf-supervised **T**ransformer with **E**nergy-based **G**raph **O**ptimization), a novel framework that distills unsupervised features into high-quality discrete semantic labels. At the core of STEGO is a novel contrastive loss function that encourages features to form compact clusters while preserving their relationships across the corpora. STEGO yields a significant improvement over the prior state of the art, on both the CocoStuff (+**14 mIoU**) and Cityscapes (+**9 mIoU**) semantic segmentation challenges.

4.3 Introduction

In Chapter 3 we showed that relationships between "global" image representations have the power to discover hidden connections in the visual arts, and "blind spots" in image generation systems. In this chapter we show that by examining relationships between "dense" or "local" image features we can automatically discover and segment visual objects from entirely unlabeled images. The key intuition behind this idea is that while the global features of Chapter 3 encode a global notion of whether two images are semantically related, the "local" features of that network tell us when two different pixels belong to a similar object. This phenomena is entirely emergent in self-supervised algorithms like DINO [86], and gives us access to something almost as good as a supervised label per pixel. In this setting we know precisely how related two pixels are, and this relationship aligns closely with human intuitions and labels. This chapter explores how we can take this signal, and "distill" it into a small collection of object categories and pixel-level annotations with respect to those categories. In effect, this chapter shows that feature relationships allow us to discover the hidden semantics of natural images entirely without a human in the loop.

Semantic segmentation is the process of classifying each individual pixel of an image into a known ontology. Although semantic segmentation models can detect and delineate objects at a much finer granularity than classification or object detection systems, these systems are hindered by the difficulties of creating labeled training data. In particular, segmenting an image can take over 100× more effort for a human annotator than classifying or drawing bounding boxes [87]. Furthermore, in complex domains such as medicine, biology, or astrophysics, ground-truth segmentation labels may be unknown, ill-defined, or require considerable domain-expertise to provide [88].

Recently, several works introduced semantic segmentation systems that could learn from weaker forms of labels such as classes, tags, bounding boxes, scribbles, or point annotations [89–92]. However, comparatively few works take up the challenge of semantic segmentation without *any* form of human supervision or motion cues. Attempts such as Independent Information Clustering (IIC) [93] and PiCIE [85] aim to learn semantically meaningful features through transformation equivariance, while imposing a clustering step to improve the compactness of the learned features.

In contrast to these previous methods, we utilize pre-trained features from unsupervised feature learning frameworks and focus on distilling them into a compact and discrete structure while preserving their relationships across the image corpora. This is motivated by the observation that correlations between unsupervised features, such as ones learned by DINO [86], are already semantically consistent, both within the same image and across image collections.

As a result, we introduce STEGO (**S**elf-supervised **T**ransformer with **E**nergy-based **G**raph **O**ptimization), which is capable of jointly discovering and segmenting objects without human supervision. STEGO distills pretrained unsupervised visual features into semantic clusters using a novel contrastive loss. STEGO dramatically improves over prior art and is a considerable step towards closing the gap with supervised segmentation systems. We include a short video detailing the work at https://aka.ms/stego-video. Specifically, we make the following contributions:

- Show that unsupervised deep network features have correlation patterns that are largely consistent with true semantic labels.

- Introduce STEGO, a novel transformer-based architecture for unsupervised semantic segmentation.

- Demonstrate that STEGO achieves state of the art performance on both the CocoStuff (+**14 mIoU**) and Cityscapes (+**9 mIoU**) segmentation challenges.

- Justify STEGO's design with an ablation study on the CocoStuff dataset.

4.4 Related Work

Self-supervised Visual Feature Learning Learning meaningful visual features without human annotations is a longstanding goal of computer vision. Approaches to this problem often optimize a surrogate task, such as denoising [94], inpainting [95], jigsaw puzzles, colorization [96], rotation prediction [97], and most recently, contrastive learning over multiple augmentations [98–101]. Contrastive learning approaches, whose performance surpass all other surrogate tasks, assume visual features are invariant under a certain set of image augmentation operations. These approaches maximize feature similarities between an image and its augmentations, while minimizing similarity between negative samples, which are usually randomly sampled images. Some notable examples of positive pairs include temporally adjacent images in videos [101], image augmentations [99, 100], and local crops of a single image [98]. Many works highlight the importance of large numbers of negative samples during training. To this end [102] propose keeping a memory bank of negative samples and [100] propose momentum updates that can efficiently simulate large negative batch sizes. Recently some works have aimed to produce spatially dense feature maps as opposed to a single global vector per image. In this vein, VADeR [103] contrasts local per-pixel features based on random compositions of image transformations that induce known correspondences among pixels which act as positive pairs for contrastive training. Instead of trying to learn visual features and clustering from scratch, STEGO treats pretrained self-supervised features as input and is agnostic to the underlying feature extractor. This makes it easy to integrate future advances in self-supervised feature learning into STEGO.

Unsupervised Semantic Segmentation Many unsupervised semantic segmentation approaches use techniques from self-supervised feature learning. IIC [93] maximizes mutual information of patch-level cluster assignments between an image and its augmentations. Contrastive Clustering [104], and SCAN [105] improve on IIC's image clustering results with supervision from negative samples and nearest neighbors but do not attempt semantic segmentation. PiCIE [85] improves on IIC's semantic segmentation results by using invariance to photometric effects and equivariance to geometric transformations as an inductive bias. In PiCIE, a network minimizes the distance between features under different transformations, where the distance is defined by an in-the-loop k-means clustering process. SegSort [106] adopts a different approach. First, SegSort learns good features using superpixels as proxy segmentation maps, then uses Expectation-Maximization to iteratively refine segments over a spherical embedding space. In a similar vein, MaskContrast [107] achieves promising results on PascalVOC by first using an off-the-shelf saliency model to generate a binary mask for each image. MaskContrast then contrasts learned features within and across the saliency masks. In contrast, our method focuses refining existing pretrained self-supervised visual features to distill their correspondence information and encourage cluster formation. This is similar to the work of [108] who show that low rank factorization of deep network features can be useful for unsupervised co-segmentation. We are not aware of any previous work that achieves the goal of high-quality, pixel-level unsupervised semantic segmentation on large scale datasets with diverse images.

Visual Transformers Convolutional neural networks (CNNs) have long been state of the art for many computer vision tasks, but the nature of the convolution operator makes it hard to model long-range interactions. To circumvent such shortcomings, [109, 110] use self-attention operations within a CNN to model long range interactions. Transformers [111], or purely self-attentive networks, have made significant progress in NLP and have recently been used for many computer vision tasks [86, 112–114]. Visual Transformers (ViT) [111] apply self-attention mechanisms to image patches and positional embeddings in order to generate features and predictions. Several modifications of ViT have been proposed to improve supervised learning, unsupervised learning, multi-scale processing, and dense predictions. In particular, DINO [86] uses a ViT within a self-supervised learning framework that performs self-distillation with exponential moving average updates. [86] show that DINO's class-attention can produce localized and semantically meaningful salient object segmentations. Our work shows that DINO's features not only detect salient objects but can be used to extract dense and semantically meaningful correspondences between images. In STEGO, we refine the features of this pre-trained backbone to yield semantic segmentation predictions when clustered. We focus on DINO's embeddings because of their quality but note that STEGO can work with any deep network features.

4.5 Methods

4.5.1 Feature Correspondences Predict Class Co-Occurrence

Recent progress in self-supervised visual feature learning has yielded methods with powerful and semantically relevant features that improve a variety of downstream tasks. Though most works aim to generate a single vector for an image, many works show that intermediate dense features are semantically relevant [108, 115, 116], a topic we will treat in theoretical detail in Chapter 7. To use this information, we focus on the "correlation volume" [117] between the dense feature maps. For convolutional or transformer architectures, these dense feature maps can be the activation map of a specific layer. Additionally, the Q, K or V matrices in transformers can also serve as candidate features, though we find these attention tensors do not perform as well in practice. More formally, let $f \in \mathbb{R}^{CHW}, g \in \mathbb{R}^{CIJ}$ be the feature tensors for two different images where C represents the channel dimension and $(H, W), (I, J)$ represent spatial dimensions. We form the feature correspondence tensor:

$$F_{hwij} := \sum_c \frac{f_{chw}}{|f_{hw}|} \frac{g_{cij}}{|g_{ij}|}, \tag{4.1}$$

whose entries represent the cosine similarity between the feature at spatial position (h, w) of feature tensor f and position (i, j) of feature tensor g. In the special case where $f = g$ these correspondences measure the similarity between two regions of the same image. We note that this quantity appears often as the "cost-volume" within the optical flow literature, and Chapter 7 will show this acts a higher-order generalization of Class Activation Maps [116] for contrastive architectures and visual search engines [115]. By examining slices of the correspondence tensor, F, at a given (h, w) we are able to visualize how two images relate according the featurizer. For example, Figure 4.2 shows how three different points from the source image (shown in blue, red, and green) are in correspondence with relevant semantic areas within the image and its K-nearest neighbors with respect to the DINO [86] as the feature extractor.

This feature correspondence tensor not only allows us to visualize image correspondences but is strongly correlated with the true label co-occurrence tensor. In particular, we can form the ground truth label co-occurrence tensor given a pair of ground-truth semantic segmentation labels $k \in \mathcal{C}^{HW}, l \in \mathcal{C}^{IJ}$ where $\mathcal{C}$ represents the set of possible classes:

$$L_{hwij} := \begin{cases} 1, & \text{if } l_{hw} = k_{ij} \\ 0, & \text{if } l_{hw} \neq k_{ij} \end{cases}$$

By examining how well the feature correspondences, F, predict the ground-truth label co-occurrences, L, we can measure how compatible the features are with the semantic

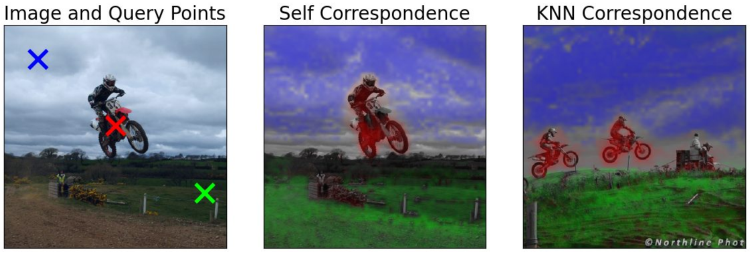


Figure 4.2: Feature correspondences from DINO. Correspondences between the source image (left) and the target images (middle and right) are plotted over the target images in the respective color of the source point (crosses in the left image). Feature correspondences can highlight key aspects of shared semantics within a single image (middle) and across similar images such as KNNs (right)

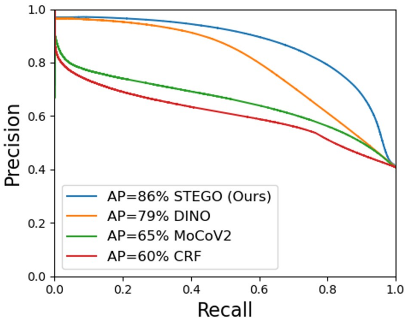


Figure 4.3: Precision recall curves show that feature self-correspondences strongly predict true label co-occurrence. DINO outperforms MoCoV2 and a CRF kernel, which shows its power as an unsupervised learning signal.

segmentation labels. More specifically we treat the feature correspondences as a probability logit and compute the average precision when used as a classifier for L. This approach not only acts as a quick diagnostic tool to determine the efficacy of features, but also allows us to compare with other forms of supervision such as the fully connected Conditional Random Field (CRF) [118], which uses correspondences between pixels to refine low-resolution label predictions. In Figure 4.3 we plot precision-recall curves for the DINO backbone, the MoCoV2 backbone, the CRF Kernel, and our trained STEGO architecture. Interestingly, we find that DINO is already a spectacular predictor of label co-occurrence within the Coco stuff dataset despite **never seeing the labels**. In particular, DINO recalls 50% of true label co-occurrences with a precision of 90% and significantly outperforms both MoCoV2 feature correspondences and the CRF kernel. One curious note is that our final trained model is a better label predictor than the supervisory signal it learns from. We attribute this to the distillation process discussed in Section 4.5.2 which amplifies this supervisory signal and drives consistency across the entire dataset. Finally, we stress that our comparison to ground truth labels within this section is solely to provide intuition about the quality of feature correspondences as a supervisory signal. **We do not use the ground truth labels to tune any parameters of STEGO.**

4.5.2 Distilling Feature Correspondences

In Section 4.5.1 we have shown that feature correspondences have the potential to be a quality learning signal for unsupervised segmentation. In this section we explore how to harness this signal to create pixel-wise embeddings that, when clustered, yield a quality semantic segmentation. In particular, we seek to learn a low-dimensional embedding that “distills” the feature correspondences. To achieve this aim, we draw inspiration from the CRF which uses an undirected graphical model to refine noisy or low-resolution class predictions by aligning them with edges and color-correlated regions in the original image.

More formally, let $\mathcal{N} : \mathbb{R}^{C'H'W'} \to \mathbb{R}^{CHW}$ represent a deep network backbone, which maps an image x with C' channels and spatial dimensions (H', W') to a feature tensor f with C channels and spatial dimensions (H, W). In this work, we keep this backbone network frozen and focus on training a light-weight segmentation head $\mathcal{S} : \mathbb{R}^{CHW} \to \mathbb{R}^{KHW}$, that maps our feature space to a code space of dimension K, where $K < C$. The goal of $\mathcal{S}$ is to learn a nonlinear projection, $\mathcal{S}(f) =: s \in \mathbb{R}^{KHW}$, that forms compact clusters and amplifies the correlation patterns of f.

To build our loss function let f and g be two feature tensors from a pair of images x, and y and let $s := \mathcal{S}(f) \in \mathbb{R}^{CHW}$ and $t := \mathcal{S}(g) \in \mathbb{R}^{CIJ}$ be their respective segmentation features. Next, using Equation 4.1 we compute a feature correlation tensor $F \in R^{HWIJ}$ from f and g and a segmentation correlation tensor $S \in R^{HWIJ}$ from s and t. Our loss function aims to push the entries of s and t together if there is a significant coupling between two corresponding entries of f and g. As shown in Figure 4.4, we can achieve this with a simple element-wise multiplication of the tensors F and S:

$$\mathcal{L}_{simple-corr}(x, y, b) := -\sum_{hwij}(F_{hwij} - b)S_{hwij} \tag{4.2}$$

Where b is a hyper-parameter which adds uniform "negative pressure" to the equation to prevent collapse. Minimizing $\mathcal{L}$ with respect to S encourages elements of S to be large when elements of $F - b$ are positive and small when elements of $F - b$ are negative. More explicitly, because the elements of F and S are cosine similarities, this exerts an attractive or repulsive force on pairs of segmentation features with strength proportional to their feature correspondences. We note that the elements of S are not just encouraged to *equal* the elements of F but rather to push to total anti-alignment (-1) or alignment (1) depending on the sign of $F - b$.

In practice, we found that $\mathcal{L}_{simple-corr}$ is sometimes unstable and does not provide enough learning signal to drive the optimization. Empirically, we found that optimizing the segmentation features towards total anti-alignment when the corresponding features do not correlate leads to instability, likely because this increases co-linearity. Therefore, we optimize weakly-correlated segmentation features to be orthogonal instead. This can be efficiently achieved by clamping the segmentation correspondence, S, at 0, which dramatically improved the optimization stability.

Additionally, we encountered challenges when balancing the learning signal for small objects which have concentrated correlation patterns. In these cases, $F_{hwij} - b$ is negative in most locations, and the loss drives the features to diverge instead of aggregate. To make the optimization more balanced, we introduce a **S**patial **C**entering operation on the feature correspondences:

$$F^{SC}_{hwij} := F_{hwij} - \frac{1}{IJ}\sum_{i'j'} F_{hwi'j'}. \tag{4.3}$$

Together with the zero clamping, our final correlation loss is defined as:

$$\mathcal{L}_{corr}(x, y, b) := -\sum_{hwij}(F^{SC}_{hwij} - b)max(S_{hwij}, 0). \tag{4.4}$$

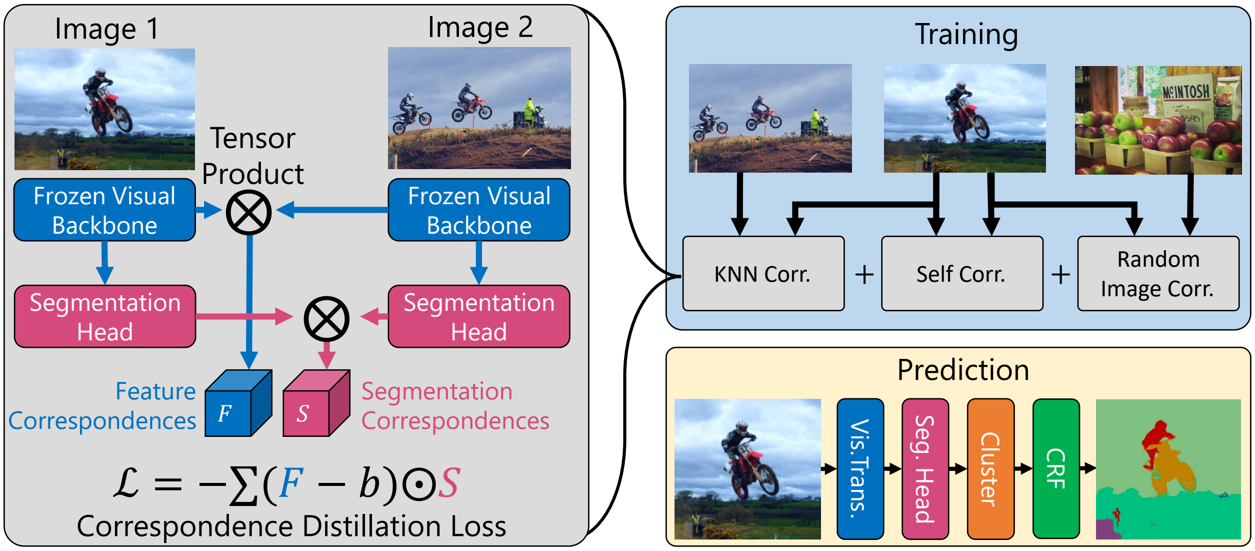


Figure 4.4: High-level overview of the STEGO architecture at train and prediction steps. Grey boxes represent three different instantiations of the main correspondence distillation loss which is used to train the segmentation head.

We demonstrate the positive effect of both the aforementioned "0-Clamp" and "SC" modifications in the ablation study of Table 4.2.

4.5.3 STEGO Architecture

STEGO uses three instantiations of the correspondence loss of Equation 4.4 to train a segmentation head to distill feature relationships between an image and itself, its K-Nearest Neighbors (KNNs), and random other images. The self and KNN correspondence losses primarily provide positive, attractive, signal and random image pairs tend to provide negative, repulsive, signal. We illustrate this and other major architecture components of STEGO in Figure 4.4.

STEGO is made up of a frozen backbone that serves as a source of learning feedback, and as an input to the segmentation head for predicting distilled features. This segmentation head is a simple feed forward network with ReLU activations [119]. In contrast to other works, our method does not re-train or fine-tune the backbone. This makes our method very efficient to train: it only takes less than 2 hours on a single NVIDIA V100 GPU card.

We first use our backbone to extract global image features by global average pooling (GAP) our spatial features: $GAP(f)$. We then construct a lookup table of each image's K-Nearest Neighbors according to cosine similarity in the backbone's feature space. Each training minibatch consists of a collection of random images x and random nearest neighbors x^{knn}. In our experiments we sample x^{knn} randomly from each image's top 7 KNNs. We also sample random images, x^{rand}, by shuffling x and ensuring that no image matched with itself. STEGO's full loss is:

$$\mathcal{L} = \lambda_{self}\mathcal{L}_{corr}(x, x, b_{self}) + \lambda_{knn}\mathcal{L}_{corr}(x, x^{knn}, b_{knn}) + \lambda_{rand}\mathcal{L}_{corr}(x, x^{rand}, b_{rand}) \tag{4.5}$$

Table 4.1: Comparison of unsupervised segmentation architectures on 27 class CocoStuff validation set. STEGO significantly outperforms prior art in both unsupervised clustering and linear-probe style metrics.

	Unsupervised		Linear Probe	
Model	**Accuracy**	**mIoU**	**Accuracy**	**mIoU**
ResNet50 [121]	24.6	8.9	41.3	10.2
MoCoV2 [100]	25.2	10.4	44.4	13.2
DINO [86]	30.5	9.6	66.8	29.4
Deep Cluster [122]	19.9	-	-	-
SIFT [123]	20.2	-	-	-
[124]	23.1	-	-	-
[125]	24.3	-	-	-
AC [126]	30.8	-	-	-
InMARS [127]	31.0	-	-	-
IIC [93]	21.8	6.7	44.5	8.4
MDC [85]	32.2	9.8	48.6	13.3
PiCIE [85]	48.1	13.8	54.2	13.9
PiCIE + H [85]	50.0	14.4	54.8	14.8
STEGO (Ours)	**56.9**	**28.2**	**76.1**	**41.0**

Where the λ's and the b's control the balance of the learning signals and the ratio of positive to negative pressure respectively. In practice, we found that a ratio of $\lambda_{self} \approx \lambda_{rand} \approx 2\lambda_{knn}$ worked well. The b parameters tended to be dataset and network specific, but we aimed to keep the system in a rough balance between positive and negative forces. More specifically we tuned the bs to keep mean KNN feature similarity at ≈ 0.3 and mean random similarity at ≈ 0.0.

Many images within the CocoStuff and Cityscapes datasets are cluttered with small objects that are hard to resolve at a feature resolution of $(40, 40)$. To better handle small objects and maintain fast training times we five-crop training images prior to learning KNNs. This not only allows the network to look at closer details of the images, but also improves the quality of the KNNs. More specifically, global image embeddings are computed for each crop. This allows the network to resolve finer details and yields five times as many images to find close matching KNNs from. Five-cropping improved both our Cityscapes results and CocoStuff segmentations, and we detail this in Table 4.2.

The final components of our architecture are the clustering and CRF refinement step. Due to the feature distillation process, STEGO's segmentation features tend to form clear clusters. We apply a cosine distance based minibatch K-Means algorithm [120] to extract these clusters and compute concrete class assignments from STEGO's continuous features. After clustering, we refine these labels with a CRF to improve their spatial resolution further.

4.5.4 Relation to Potts Models and Energy-based Graph Optimization

Equation 4.4 can be viewed in the context of Potts models or continuous Ising models from statistical physics [128, 129]. We briefly overview this connection, and point interested readers to Section B.8 for a more detailed discussion. To build the general Ising model, let $\mathcal{G} = (\mathcal{V}, w)$ be a fully connected, weighted, and undirected graph on $|\mathcal{V}|$ vertices. In our applications we take $\mathcal{V}$ to be the set of pixels in the training dataset. Let $w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ represent an edge weighting function. Let $\phi : \mathcal{V} \to \mathcal{C}$ be a vertex valued function mapping into a generic code space $\mathcal{C}$ such as the probability simplex over cluster labels $\mathcal{P}(L)$, or the K-dimensional continuous feature space $\mathbb{R}^K$. The function ϕ can be a parameterized neural network, or a simple lookup table that assigns a code to each graph node. Finally, we define a compatibility function $\mu : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ that measures the cost of comparing two codes. We can now define the following graph energy functional:

$$E(\phi) := \sum_{v_i, v_j \in \mathcal{V}} w(v_i, v_j)\mu(\phi(v_i), \phi(v_j)) \tag{4.6}$$

Constructing the Boltzmann Distribution [130] yields a normalized distribution over the function space Φ:

$$p(\phi|w, \mu) = \frac{\exp(-E(\phi))}{\int_\Phi \exp(-E(\phi'))d\phi'} \tag{4.7}$$

In general, sampling from this probability distribution is difficult because of the often-intractable normalization factor. However, it is easier to compute the maximum likelihood estimate (MLE), $\operatorname{argmax}_{\phi\in\Phi} p(\phi|w, \mu)$. In particular, if Φ is a smoothly parameterized space of functions and ϕ and μ are differentiable functions, one can compute the MLE using stochastic gradient descent (SGD) with highly-optimized automatic differentiation frameworks [131, 132]. In Section B.8 of the supplement we prove that the finding the MLE of Equation 4.7 is equivalent to minimizing the loss of Equation 4.4 when $|V|$ is the set of pixels in our image training set, $\phi = \mathcal{S} \circ \mathcal{N}$, w is the cosine distance between features, and μ is cosine distance. Like STEGO, the CRF is also a Potts model, and we use this connection to re-purpose the STEGO loss function to create continuous, minibatch, and unsupervised variants of the CRF. We detail this exploration in Section B.9 of the Supplement.

4.6 Experiments

We evaluate STEGO on standard semantic segmentation datasets and compare with current state-of-the-art. We then justify different design choices of STEGO through ablation studies. Additional details on datasets, model hyperparameters, hardware, and other implementation details can be found in Section B.10 of the Supplement.

4.6.1 Evaluation Details

Datasets Following [85], we evaluate STEGO on the 27 mid-level classes of the CocoStuff class hierarchy and on the 27 classes of Cityscapes. Like prior art, we first resize images to

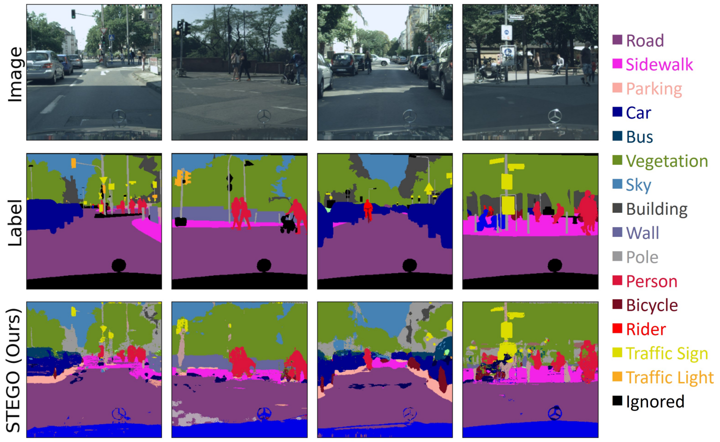


Figure 4.5: Comparison of ground truth labels (middle row) and cluster probe predictions for STEGO (bottom row) for images from the Cityscapes dataset.

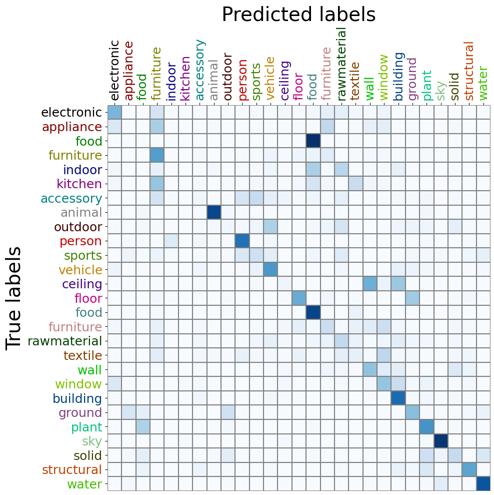


Figure 4.6: Confusion matrix of STEGO cluster probe predictions on CocoStuff. Classes after the "vehicle" class are "stuff" and classes before are "things". Rows are normalized to sum to 1.

320 pixels along the minor axis followed by a (320 × 320) center crops of each validation image. We use mean intersection over union (mIoU) and Accuracy for evaluation metrics. Our CocoStuff evaluation setting originated in [93] and is common in the literature. Our Cityscapes evaluation setting is adopted from [85]. The latter is newer and more challenging, and thus fewer baselines are available. Finally we also compare on the Potsdam-3 setting fro [93] in Section B.2 of the Appendix.

Linear Probe The first way we evaluate the quality of the distilled segmentation features is through transfer learning effectiveness. As in [85, 107, 133], we train a linear projection from segmentation features to class labels using the cross entropy loss. This loss solely evaluates feature quality and is not part of the STEGO training process.

Clustering Unlike the linear probe, the clustering step does not have access to ground truth supervised labels. As in prior art, we use a Hungarian matching algorithm to align our unlabeled clusters and the ground truth labels for evaluation and visualization purposes. This measures how consistent the predicted semantic segments are with the ground truth labels and is invariant to permutations of the predicted class labels.

4.6.2 Results

We summarize our main results on the 27 classes of CocoStuff in Table 4.1. STEGO significantly outperforms the prior state of the art, PiCIE, on both linear probe and clustering (Unsupervised) metrics. In particular, STEGO improves by +**14** unsupervised mIoU, +**6.9** unsupervised accuracy, +**26** linear probe mIoU, and +**21** linear probe accuracy compared to the next best baseline. In Table 4.3, we find a similarly large improvement of +**8.7** unsupervised mIoU and +**7.7** unsupervised accuracy on the Cityscapes validation set. These two experiments demonstrate that even though we do not fine-tune the backbone for these datasets, DINO's self-supervised weights on ImageNet [134] are enough to simultaneously solve both settings. STEGO also outperforms simply clustering the features from unmodified DINO, MoCoV2, and ImageNet supervised ResNet50 backbones. This demonstrates the benefits of training a segmentation head to distill feature correspondences.

We show some example segmentations from STEGO and our baseline PiCIE on the CocoStuff dataset in Figure 4.1. We include additional examples and failure cases in Sections B.4 and B.5. We note that STEGO is significantly better at resolving fine-grained details within the images such as the legs of horses in the third image from the left column of Figure 4.1, and the individual birds in the right-most column. Though the PiCIE baseline uses a feature pyramid network to output high resolution predictions, the network does not attune to fine grained details, potentially demonstrating the limitations of the sparse training signal induced by data augmentations alone. In contrast, STEGO's predictions capture small objects and fine details. In part, this can be attributed to DINO backbone's higher resolution features, the 5-crop training described in 4.5.3, and the CRF post-processing which helps to align the predictions to image edges. We show qualitative results on the Cityscapes dataset in Figure 4.5. STEGO successfully identifies people, street, sidewalk, cars, and street signs with high detail and fidelity. We note that prior works did not publish pretrained models or linear probe results on Cityscapes so we exclude this information from Table 4.3 and Figure 4.5.

To better understand the predictions and failures of STEGO, we include confusion matrices for CocoStuff (Figure 4.6) and Cityscapes (Figure B.5 of the Supplement). Some salient STEGO errors include confusing the "food" category from the CocoStuff "things", and the "food" category from CocoStuff "stuff". STEGO also does not properly separate "ceilings" from "walls", and lacks consistent segmentations for classes such as "indoor", "accessory", "rawmaterial" and "textile". These errors also draw our attention to the challenges of evaluating unsupervised segmentation methods: *label ontologies can be arbitrary*. In these circumstances the divisions between classes are not well defined and it is hard to imagine a system that can segment the results consistently without additional information. In these regimes, the linear probe provides a more important barometer for quality because the limited supervision can help disambiguate these cases. Nevertheless, we feel that there is still considerable progress to be made on the purely unsupervised benchmark, and that even with the improvements of STEGO there is still a measurable performance gap with supervised systems.

4.6.3 Ablation Study

To understand the impact of STEGO's architectural components we perform an ablation analysis on the CocoStuff dataset, and report the results in Table 4.2. We examine the effect of using several different backbones in STEGO including MoCoV2, the ViT-Small, and ViT-Base architectures of DINO. We find that ViT-Base is the best feature extractor of the group and leads by a significant margin both in terms of accuracy and mIoU. We also evaluate the several loss function and architecture decisions described in Section 4.5.3. In particular, we explore clamping the segmentation feature correspondence tensor at 0 to prevent the

Table 4.2: Architecture ablation study on the CocoStuff Dataset (27 Classes).

Arch.	0-Clamp	5-Crop	SC	CRF	Unsup. Acc.	Unsup. mIoU	Linear Probe Acc.	Linear Probe mIoU
MoCoV2	✓				48.4	20.8	70.7	26.5
ViT-S					34.2	7.3	54.9	15.6
ViT-S	✓				44.3	21.3	70.9	36.8
ViT-S	✓	✓			47.6	23.4	72.2	36.8
ViT-S	✓	✓	✓		47.7	24.0	72.9	38.4
ViT-S	✓	✓	✓	✓	48.3	24.5	74.4	38.3
ViT-B	✓	✓	✓		54.8	26.8	74.3	39.5
ViT-B	✓	✓	✓	✓	**56.9**	**28.2**	**76.1**	**41.0**

Table 4.3: Results on the Cityscapes Dataset (27 Classes). STEGO improves significantly over all baselines in both accuracy and mIoU.

Model	Unsup. Acc.	Unsup. mIoU
IIC [93]	47.9	6.4
MDC [85]	40.7	7.1
PiCIE [85]	65.5	12.3
STEGO (Ours)	**73.2**	**21.0**

negative pressure from introducing co-linearity (0-Clamp), five-cropping the dataset prior to mining KNNs to improve the resolution of the learning signal (5-Crop), spatially centering the feature correspondence tensor to improve resolution of small objects (SC), and Conditional Random Field post-processing to refine predictions (CRF). We find that these modifications improve both the cluster and linear probe evaluation metrics.

4.7 Chapter Conclusion

We have found that modern self-supervised visual backbones can be refined to yield state of the art unsupervised semantic segmentation methods. We have motivated this architecture by showing that correspondences between deep features are directly correlated with ground truth label co-occurrence. We take advantage of this strong, yet entirely unsupervised, learning signal by introducing a novel contrastive loss that "distills" the correspondences between features. Our system, STEGO, produces low rank representations that cluster into accurate semantic segmentation predictions. We connect STEGO's loss to CRF inference by showing it is equivalent to MLE in Potts models over the entire collection of pixels in our dataset. We show STEGO yields a significant improvement over the prior state of the art, on both the CocoStuff (+**14 mIoU**) and Cityscapes (+**9 mIoU**) semantic segmentation challenges. Finally, we justify the architectural decisions of STEGO with an ablation study on the CocoStuff dataset.

Chapter 5

FeatUp: Improving the Resolution of any Model by Upsampling Visual Representations

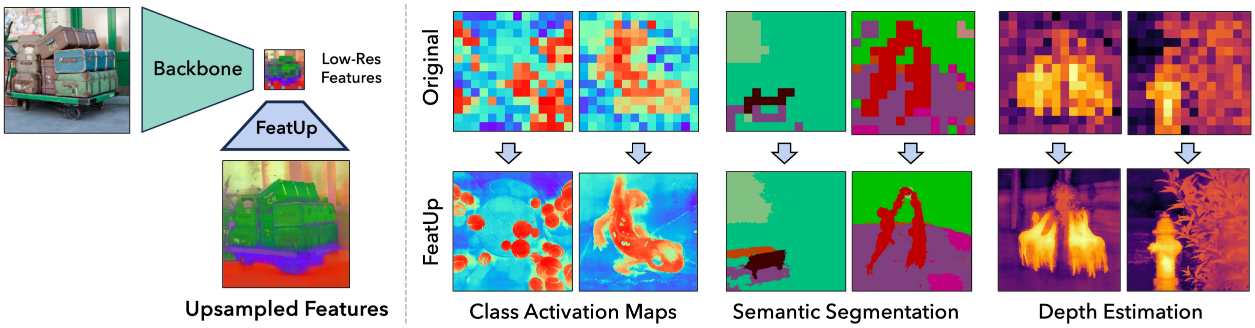


Figure 5.1: FeatUp upsamples image features from any model backbone, adding spatial resolution to existing semantics. High-res features can be learned either as a per-image implicit network or a general-purpose upsampling operation; the latter is a drop-in module to improve downstream dense prediction tasks.

5.1 Website and Video

For a quick video overview and blog post of this chapter, see https://mhamilton.net/featup.html

5.2 Chapter Summary

Chapters 3 and 4 present two applications that demonstrate the hidden power of deep features to capture detailed semantics about our world without any reliance on human labels. More generally, these deep features are a cornerstone of computer vision research, capturing image semantics and enabling the community to solve downstream tasks even in the zero- or few-shot

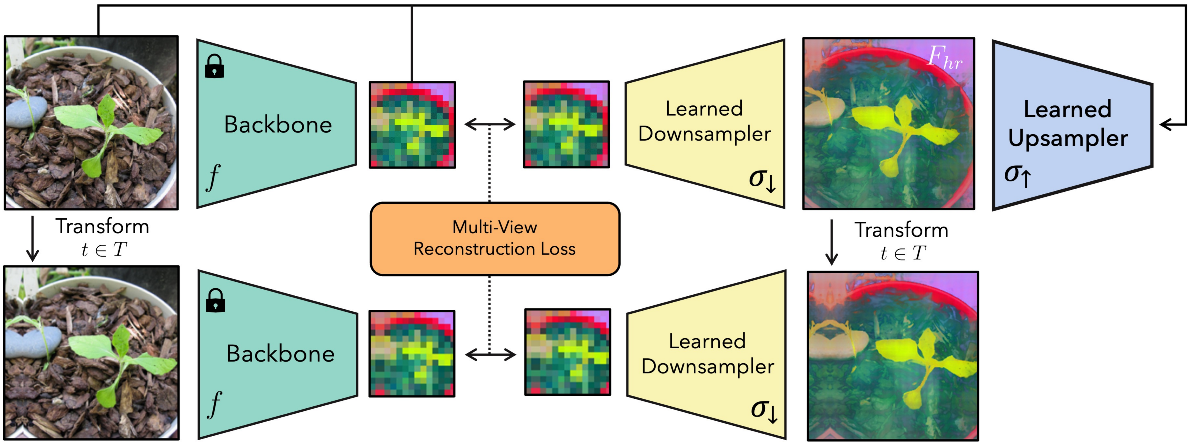


Figure 5.2: The FeatUp training architecture. FeatUp learns to upsample features through a consistency loss on low resolution "views" of a model's features that arise from slight transformations of the input image.

regime. However, these features often lack the spatial resolution to directly perform dense prediction tasks like segmentation and depth prediction because models aggressively pool information over large areas. In this work, we introduce FeatUp, a task- and model-agnostic framework to restore lost spatial information in deep features. We introduce two variants of FeatUp: one that guides features with high-resolution signal in a single forward pass, and one that fits an implicit model to a single image to reconstruct features at any resolution. Both approaches are powered by analyzing how slight variations in an input image cause deep representations to subtly change. In particular FeatUp uses a multi-view consistency loss with deep analogies to NeRFs, to recover this missing spatial information without human supervision. Our features retain their original semantics and can be swapped into existing applications to yield resolution and performance gains even without re-training. We show that FeatUp significantly outperforms other feature upsampling and image super-resolution approaches in class activation map generation, transfer learning for segmentation and depth prediction, and end-to-end training for semantic segmentation. Additionally, FeatUp can be combined with the STEGO from Chapter 4 to dramatically improve the resolution and quality of unsupervised semantic segmentation.

5.3 Introduction

Chapters 3 and 4 both show that relationships between deep features are the key to discovering semantics from totally unlabeled datasets. More broadly in the community, considerable effort has been made to develop methods to extract features from data modalities such as vision [86, 123, 135–137], text [138–140], and audio [141, 142]. These features often form the backbone of different methods, including classification [143], weakly-supervised learning [144, 145], semantic segmentation [146], optical flow [117, 147], neural rendering [148], and more recently, image generation [149]. Despite their immense success, deep features often sacrifice spatial resolution for semantic quality. For example, ResNet-50 [150] produces 7 × 7 deep features from a 224 × 224 pixel input (32× resolution reduction). Even Vision Transformers (ViTs) [151] incur a significant resolution reduction, making it challenging to perform dense prediction tasks such as segmentation or depth estimation using these features alone.

To mitigate these issues, this chapter proposes FeatUp: a novel framework to improve the resolution of any vision model's features without changing their original "meaning" or orientation. Our primary insight, inspired by 3D reconstruction frameworks like NeRF [152], is that multiview consistency of low-resolution signals can supervise the construction of high-resolution signals. More specifically, we learn high-resolution information by aggregating low resolution views from a model's outputs across multiple "jittered" (e.g. flipped, padded, cropped) images. We aggregate this information by learning an upsampling network with a multiview consistency loss. To put this in the broader context of this thesis: by analyzing how deep representations change as we translate the input image we can upsample these representations by up to 64× without any supervision. This not just improves the algorithms of Chapter 4 and Chapter 6 to come, but can improve *any* algorithm built using deep representations.

This chapter explores two architectures for upsampling: a single guided upsampling feedforward network that generalizes across images, and an implicit representation overfit to a single image. This feedforward upsampler is a parameterized generalization of a Joint Bilateral Upsampling (JBU) filter [153] powered by a CUDA kernel orders of magnitude faster and less memory-intensive than existing implementations. This upsampler can produce high quality features aligned to object edges at a computational cost comparable to a few convolutions. Our implicit upsampler draws a direct parallel to NeRF and overfits a deep implicit network to a signal, allowing for arbitrary resolution features and low storage costs. In both architectures, our upsampled features can be drop-in replacements in downstream applications because our methods do not transform the semantics of the underlying features. We show that these upsampled features can significantly improve a variety of downstream tasks including semantic segmentation and depth prediction. Additionally, we show that model explanation methods such as CAM can be made higher-resolution using upsampled features. In particular, one can study a model's behavior with much greater detail without the need for complex methods based on relevance and information propagation [154, 155]. In summary, we include a short video describing FeatUp at aka.ms/featup and make the following contributions:

- FeatUp: a new method to significantly improve the spatial resolution of any model's features, parametrized as either a fast feedforward upsampling network or an implicit network.

- A fast CUDA implementation of Joint Bilateral Upsampling orders of magnitude more efficient than a standard PyTorch implementation and allowing guided upsampling in large-scale models.

- We show that FeatUp features can be used as drop-in replacements for ordinary features to improve performance on dense prediction tasks and model explainability.

5.4 Related Work

Image-adaptive filtering. Adaptive filters are commonly used to enhance images while preserving their underlying structure and content. For example, bilateral filters [156–158] apply a spatial filter to a low-resolution signal and an intensity filter to a high-resolution guidance to blend information from the two. Joint Bilateral Upsampling (JBU) [153] uses this technique to upsample a low-resolution signal with a high-resolution guidance. JBU has been used successfully for efficient image enhancement and other applications. Recently, some works embed bilateral filtering approaches [159] and nonlocal means [160] into convolutional networks [161–164] and vision transformers [165, 166]. Shape Recipes [167] learn the local relationship between signals to create up-sample target signals. Pixel-adaptive convolutional (PAC) networks [168] adapt a convolution operation to input data and has been used to advance performance in segmentation [169, 170] and monocular depth estimation [171–173]. The Spatially-Adaptive Convolution (SAC) in [174] factorizes the adaptive filter into an attention map and convolution kernel. [175] extend bilateral filtering to superpixels and embed this operation inside of a deep network to improve semantic segmentation. This class of methods, effective across a variety of applications, directly incorporates spatial information into the task while still allowing for flexibility in learning a network.

Image super-resolution. One of the earliest deep unsupervised super-resolution methods was Zero-Shot Super-resolution (ZSSR) [176], which learns a single-image network at test time. Local implicit models [177] use locally-adaptive models to interpolate information, and have been shown to improve the performance of super-resolution networks. Deep Image Priors [178] show that CNNs provide inductive biases for inverse problems such as zero-shot image denoising and super-resolution. While there is extensive literature on image super-resolution, these methods are not well-adapted to handle ultra-low resolution, yet high-dimensional deep features as we show in the Supplement.

General-purpose feature upsampling. A widely-used approach to upsample deep feature maps is bilinear interpolation. Though efficient, this method blurs information and is insensitive to the content or the high-resolution structure in the original image. Nearest neighbor and bicubic interpolation [179] have similar drawbacks. Evaluating a network on larger inputs can achieve higher resolutions but with a steep computational cost. Furthermore, this often degrades model performance and semantics due to the decreased relative receptive field size. For deep convolutional networks, one popular technique is to set final convolution strides to 1 [155, 180]. However, this approach yields blurry features, as the model's receptive field is still large. Recent works using visual transformers [181, 182] perform a similar modification on input patch strides and interpolate positional encodings. Though simple and reasonably effective, this approach incurs a steep increase in computational footprint for every 2× increase in resolution, making it impossible to use in practice for larger upsampling factors. This approach can also distort features because of the previously mentioned fixed receptive field of the patches.

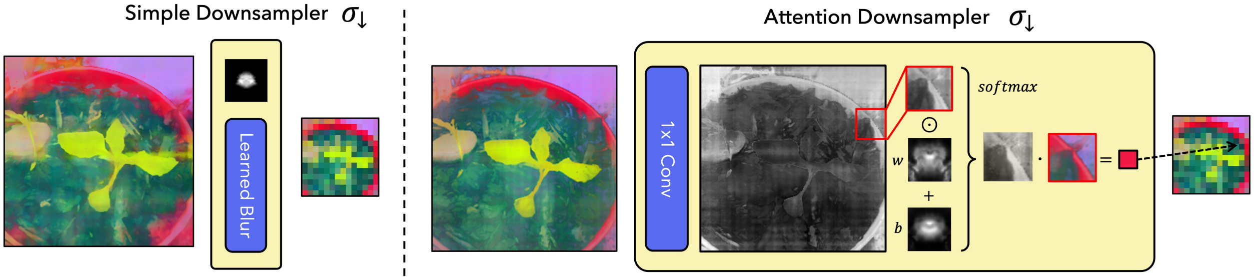

Simple Downsampler $\sigma_\downarrow$
Learned Blur
Attention Downsampler $\sigma_\downarrow$
1x1 Conv
softmax
⊙
w
+
b
=

Figure 5.3: We introduce two learned downsamplers. The simple downsampler (Left) is a fast learned blur kernel. The attention downsampler (right) combines a predicted salience map with spatially invariant kernels. This downsampler can better adapt to networks with nonlinear and dynamic receptive fields.

Image-adaptive feature upsampling. Many different operations exist in the literature to create features at higher resolutions. Deconvolutions [183–186] and transposed convolutions [187] use a learned kernel to transform features into a new space with a larger resolution. The resize-convolution [188] appends a learned convolution to deterministic upsampling procedure and reduces checkerboard artifacts that plague deconvolutions [188–190]. The resize-convolution is now a common component of image decoders such as the U-Net [191] and has been applied to semantic segmentation [192–194] and super-resolution [195–197]. Other methods such as IndexNet [198] and Affinity-Aware Upsampling (A2U) [199] are effective on image matting but fall short on other dense prediction tasks [200]. Methods such as Pixel-Adaptive Convolutions [168], CARAFE [201]SAPA [202], and DGF [203] use learned input-adaptive operators to transform features. Though PAC is flexible, it does not upsample *existing* feature maps faithfully and instead is used to transform features for downstream tasks. Additionally, DGF approximates the JBU operation with learned pointwise convolutions and linear maps, but does not fully implement JBU because the local query/model is computationally intractable. This is precisely the problem we solve exactly with our new efficient CUDA kernel. Additionally, FADE [200] introduces a new semi-shift operator and uses decoder features to produce a joint feature upsampling module. [204] view feature upsampling in a different light, focusing on a nearest-neighbors approach to align feature maps in encoder-decoder architectures with IFA. While IFA performs well on the specific semantic segmentation benchmarks, it does not take advantage of image guidance and fails to learn high quality representations outside of the encode-decoder framework, as we show in the Supplement.

5.5 Methods

The core intuition behind FeatUp is that one can compute high-resolution features by observing multiple different "views" of low-resolution features. We draw a comparison with 3D scene reconstruction models such as NeRF [152]; in the same way that NeRF builds an implicit representation [205, 206] of a 3D scene by enforcing consistency across many 2D photos of the scene, FeatUp builds an upsampler by enforcing consistency across many

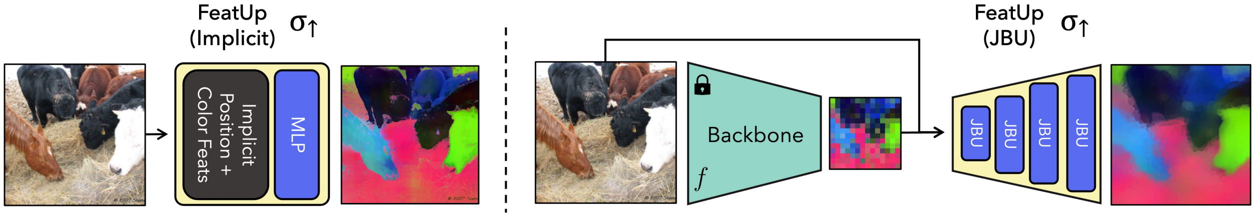


Figure 5.4: Our Implicit version of FeatUp learns an implicit network to upsample a single image's features. Our JBU FeatUp learns a stack of JBUs that learns to quickly upsample features from a large image corpora.

low-resolution feature maps. Like in broader NeRF literature, a variety of methods can arise from this basic idea. In this work, we introduce a lightweight, forward-pass upsampler based on Joint Bilateral Upsampling [153] as well as an implicit network based upsampling strategy. The latter is learned per-image and query-able at arbitrary resolution. We provide an overview of the general FeatUp architecture in Figure 5.2.

The first step in our pipeline is to generate low-resolution feature views to refine into a single high-resolution output. To this end, we perturb the input image with small pads, scales, and horizontal flips and apply the model to each transformed image to extract a collection of low-resolution feature maps. These small image jitters allow us to observe tiny differences in the output features and provide sub-feature information to train the upsampler.

Next, we construct a consistent high-resolution feature map from these views. We postulate that we can learn a latent high-resolution feature map that, when downsampled, reproduces our low-resolution jittered features (see Figure 5.2). FeatUp's downsampling is a direct analog to ray-marching; just as 3D data is rendered into 2D in this NeRF step, our downsampler transforms high-resolution features into low-resolution features. Unlike NeRF, we do not need to estimate parameters that generate each view. Instead, we track the parameters used to "jitter" each image and apply *the same* transformation to our learned high-resolution features prior to downsampling. We then compare downsampled features to the true model outputs using a gaussian likelihood loss [207]. A good high-resolution feature map should reconstruct the observed features across all the different views.

More formally, let $t \in T$ be from a collection of small transforms such as pads, zooms, crops, horizontal flips, and their compositions. Let x be an input image, f be our model backbone, $\sigma_\downarrow$ be a learned downsampler, and $\sigma_\uparrow$ be a learned upsampler. We can form the predicted high-res features F_{hr} by evaluating $F_{hr} = \sigma_\uparrow(f(x), x)$. We note that this parameterization allows $\sigma_\uparrow$ to be a guided upsampler (which depends on both x and $f(x)$), an unguided upsampler (which depends on only $f(x)$), an implicit network (which depends on only x), or a learned buffer of features (which depends on nothing). We can now form our main multi-view reconstruction loss term as follows:

$$\mathcal{L}_{rec} = \frac{1}{|T|} \sum_{t \in T} \frac{1}{2s^2} \| f\left(t\left(x\right)\right) - \sigma_\downarrow\left(t\left(F_{hr}\right)\right) \|_2^2 + \log(s) \tag{5.1}$$

Where $\|\cdot\|$ is the standard squared l_2 norm and $s = \mathcal{N}(f\left(t\left(x\right)\right))$ is a spatially-varying adaptive uncertainty [207] parameterized by a small linear network $\mathcal{N}$. This turns the MSE loss into a proper likelihood capable of handling uncertainty. This extra flexibility allows

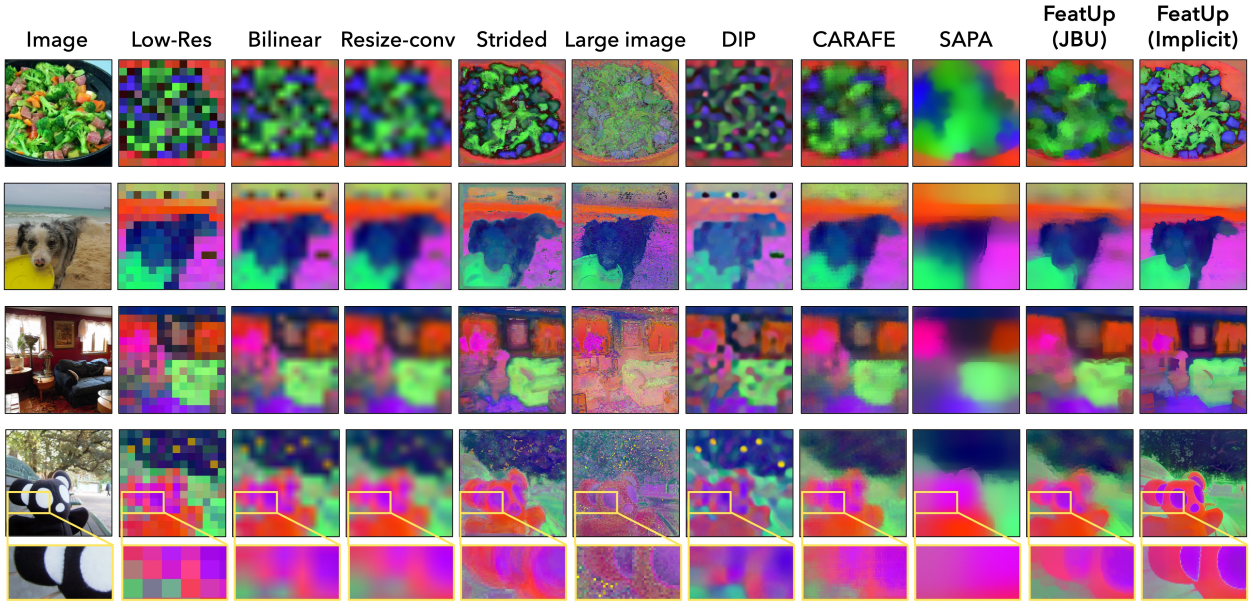


Figure 5.5: Low-res ViT features (14×14) from the COCO-Stuff validation set are upsampled by 16×. Bilinear and resize-conv baselines produce blurry outputs. Larger inputs and smaller transformer strides can help, but introduce noise or blur and are bound by time and memory constraints (We can only compute 8× upsamplings for these methods, see Figure C.10). Our FeatUp methods preserve semantics of the low-res features and recover lost spatial information from the high-res input image.

the network to learn when certain outlier features fundamentally cannot be upsampled. In the supplement, we show this adaptive uncertainty's effectiveness in an ablation study and visualization.

5.5.1 Choosing a Downsampler

Our next architectural choice is the learned downsampler $\sigma_{\downarrow}$. We introduce two options: a fast and simple learned blur kernel, and a more flexible attention-based downsampler. Both proposed modules do not change the "space" or "semantics" of the features with nontrivial transformations, but rather only interpolate features within a small neighborhood. We diagram both choices in Figure 5.3 and demonstrate the effectiveness of the attention downsampler in Figure C.2 of the Supplement.

Our simple downsampler blurs the features with a learned blur kernel and can be implemented as a convolution applied independently to each channel. The learned kernel is normalized to be non-negative and sum to 1 to ensure the features remain in the same space.

Though this blur-based downsampler is efficient, it cannot capture dynamic receptive fields, object salience, or other nonlinear effects. To this end, we also introduce a more flexible attention downsampler that spatially adapts the downsampling kernel. In short, this component uses a 1x1 convolution to predict a saliency map from the high-resolution features. It combines this saliency map with learned spatially-invariant weight and bias kernels and normalizes the result to create a spatially-varying blur kernel that interpolates the features.

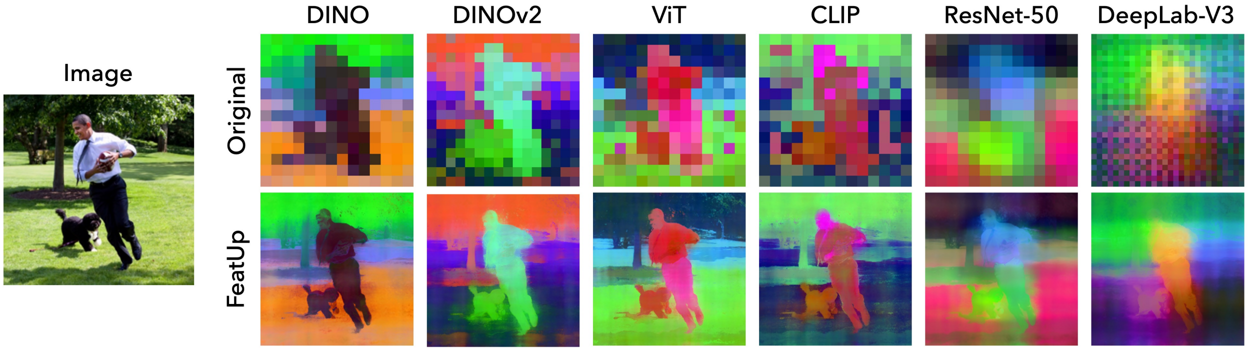


Figure 5.6: FeatUp can upsample the features of any backbone, even convnets with aggressive nonlinear pooling.

More formally:

$$\sigma_{\downarrow}(F_{hr})_{ij} = \text{softmax}(w \odot \text{Conv}(F_{hr}[\Omega_{ij}]) + b) \cdot F_{hr}[\Omega_{ij}] \tag{5.2}$$

Where $\sigma_{\downarrow}(F)_{ij}$ is the i, jth component of the resulting feature map and $F_{hr}[\Omega_{ij}]$ refers to a patch of high resolution features corresponding to the i, j location in the downsampled features. $\odot$ and $\cdot$ refer to the elementwise and inner products respectively, and w and b are learned weight and bias kernels shared across all patches. Our main hyperparameter for both downsamplers is the kernel size, which should be larger for models with larger receptive fields such as convolutional nets. We defer discussion of model-specific hyperparameters to the Supplement.

5.5.2 Choosing an Upsampler

A central choice in our architecture is the parameterization of $\sigma_{\uparrow}$. We introduce two variants: "JBU" FeatUp parameterizes $\sigma_{\uparrow}$ with a guided upsampler based on a stack of Joint Bilateral Upsamplers (JBU) [153]. This architecture learns an upsampling strategy that generalizes across a corpus of images. The second method, "Implicit" FeatUp, uses an implicit network to parameterize $\sigma_{\uparrow}$ and can yield remarkably crisp features when overfit to a single image. Both methods are trained using the same broader architecture and loss. We illustrate both strategies in Figure 5.4.

Joint Bilateral Upsampler. Our feedforward upsampler uses a stack of parameterized joint bilateral upsamplers (JBU) [153]:

$$F_{hr} = (\text{JBU}(\cdot, x) \circ \text{JBU}(\cdot, x) \circ ...)(f(x)) \tag{5.3}$$

where $\circ$ is function composition, $f(x)$ is the low-resolution feature map, and x is the original image. This architecture is fast, directly incorporates high-frequency details from the input image into the upsampling process, and is independent of the architecture of f. Our formulation generalizes the original JBU [153] implementation to high-dimensional signals and makes this operation learnable. In joint bilateral upsampling we use a high-resolution signal, G, as guidance for the low-resolution features F_{lr}. We let Ω be a neighborhood of each pixel in the guidance. In practice, we use a 3×3 square centered at each pixel. Let $k(\cdot, \cdot)$ be a similarity kernel that measures how "close" two vectors are. We can then form our joint bilateral filter:

$$\hat{F}_{hr}[i,j] = \frac{1}{Z} \sum_{(a,b)\in\Omega} \Big(F_{lr}[a,b] \;\; k_{range}(G[i,j], G[a,b]) \;\; k_{spatial}([i,j],[a,b]) \Big) \tag{5.4}$$

where Z is a normalization factor to ensure the kernel sums to 1. Here, $k_{spatial}$ is a learnable Gaussian kernel on the Euclidean distance between coordinate vectors of width $\sigma_{spatial}$:

$$k_{spatial}(x,y) = \exp\left(\frac{-\|x-y\|_2^2}{2\sigma_{spatial}^2} \right) \tag{5.5}$$

Furthermore, k_{range} is a temperature-weighted softmax [207] applied to the inner products from a multi-layer perceptron (MLP) that operates on the guidance signal G:

$$k_{range}(x,y) = \text{softmax}_{(a,b)\in\Omega}\left(\frac{1}{\sigma_{range}^2} MLP(G[i,j]) \cdot MLP(G[a,b]) \right) \tag{5.6}$$

where σ_{range}^2 acts as the temperature. We note that the original JBU uses a fixed Gaussian kernel on the guidance signal, G. Our generalization performs much better as the MLP can be learned from data to create a better upsampler. In our experiments, we use a two-layer GeLU [208] MLP with 30-dimensional hidden and output vectors. To evaluate $F_{lr}[a,b]$ we follow the original JBU formulation and use bilinear-interpolated features if the guidance pixel does not directly align with a low-resolution feature. For resolution independence, we use coordinate distances normalized to $[-1,1]$ in the spatial kernel.

One challenge we faced was the poor speed and memory performance of existing JBU implementations. This could explain why this simple approach is not used more widely. To this end, we contribute an efficient CUDA implementation of the spatially adaptive kernel used in the JBU. Compared to a naive PyTorch implementation with the `torch.nn.Unfold` operator, our operation uses up to two orders of magnitude less memory and speeds inference by up to 10×. We demonstrate its significant performance improvements in Table C.4 of the supplement.

Implicit Our second upsampler architecture draws a direct analogy with NeRF by parametrizing the high-resolution features of a single image with an implicit function $F_{hr} = \text{MLP}(z)$. Several existing upsampling solutions also take this inference-time training approach, including DIP [178] and LIIF [177]. We use a small MLP to map image coordinates and intensities to a high-dimensional feature for the given location. We follow the guidance of prior works [152, 209, 210] and use Fourier features to improve the spatial resolution of our implicit representations. In addition to standard Fourier positional features, we show that adding Fourier color features allows the network to use high-frequency color information from the original image. This significantly speeds convergence and enables graceful use of high-resolution image information without techniques like Conditional Random Fields (CRFs). We illustrate the profound effect of Fourier color features in Section C.4 of the Supplement.

More formally, let $h(z, \hat{\omega})$ represent the component-wise discrete Fourier transform of an input signal z, with a vector of frequencies $\hat{\omega}$. Let e_i and e_j represent the two-dimensional pixel coordinate fields ranging in the interval $[-1, 1]$. Let $:$ represent concatenation along the channel dimension. We can now express our high-resolution feature map as:

$$F_{hr} = \text{MLP}(h(e_i : e_j : x, \hat{\omega})) \tag{5.7}$$

Our MLP is a small 3-layer ReLU [211] network with dropout [212]($p = .1$) and layer normalization [213]. We note that, at test time, we can query the pixel coordinate field to yield features F_{hr} at **any** resolution. The number of parameters in our implicit representation is over two orders of magnitude smaller than a (224×224) explicit representation while being more expressive, significantly reducing convergence time and storage size.

5.5.3 Additional Method Details

Accelerated Training with Feature Compression To reduce the memory footprint and further speed up the training of FeatUp's implicit network, we first compress the spatially-varying features to their top $k = 128$ principal components. This operation is approximately lossless as the top 128 components explain $\sim 96\%$ of the variance across a single image's features. This improves training time by a factor of 60× for ResNet-50, reduces the memory footprint, enables larger batches, and does not have any observable effect on learned feature quality. When training the JBU upsampler, we sample random projection matrices in each batch to avoid computing PCA in the inner loop. This achieves the same effect thanks to the Johnson–Lindenstrauss lemma [214].

Total Variation Prior To avoid spurious noise in the high resolution features, we add a small ($\lambda_{tv} = 0.05$) total variation smoothness prior [215] on the implicit feature magnitudes:

$$\mathcal{L}_{tv} = \sum_{i,j} \Big((||F_{hr}[i, j]|| - ||F_{hr}[i - 1, j]||)^2 + (||F_{hr}[i, j]|| - ||F_{hr}[i, j - 1]||)^2 \Big) \tag{5.8}$$

This is faster than regularizing full features and avoids overprescribing how the individual components should organize. We do not use this in the JBU upsampler because it does not suffer from overfitting. We demonstrate the importance of this regularizer in Section C.4 in the supplement.

5.6 Experiments

We compare our method against several key upsampling baselines from the literature, in particular: Bilinear upsampling, Resize-conv, Strided (i.e. reducing the stride of the backbone's patch extractor), Large Image (i.e. using a larger input image), CARAFE [201], SAPA [202], and FADE [200]. We upsample ViT [151] features by 16× (to the resolution of the input image) with every method except the strided and large-image baselines, which are computationally infeasible above 8× upsampling. For additional details on the strided implementation, please refer to Section C.2 of the Supplement.

	CAM Score		Semantic Seg.		Depth Estimation	
	A.D. ↓	A.I. ↑	Acc. ↑	mIoU ↑	RMSE ↓	$\delta > 1.25$ ↑
Low-res	10.69	4.81	65.17	40.65	1.25	0.894
Bilinear	10.24	4.91	66.95	42.40	1.19	0.910
Resize-conv	11.02	4.95	67.72	42.95	1.14	0.917
DIP	10.57	5.16	63.78	39.86	1.19	0.907
Strided	11.48	4.97	64.44	40.54	2.62	0.900
Large image	13.66	3.95	58.98	36.44	2.33	0.896
CARAFE	10.24	4.96	67.1	42.39	$\underline{1.09}$	0.920
SAPA	10.62	4.85	65.69	41.17	1.19	0.917
FeatUp (JBU)	$\underline{9.83}$	$\underline{5.24}$	$\underline{68.77}$	$\underline{43.41}$	$\underline{1.09}$	**0.938**
FeatUp (Implicit)	**8.84**	**5.60**	**71.58**	**47.37**	**1.04**	$\underline{0.927}$

Table 5.1: Comparison of feature upsamplers across metrics on CAM faithfulness, linear probe semantic segmentation, and linear probe depth estimation. Both FeatUp variants consistently outperform other approaches, including other forward-pass upsamplers (CARAFE, SAPA) and features optimized at inference-time (DIP).

5.6.1 Qualitative Comparisons

Visualizing upsampling methods Figure 5.5 demonstrates the dramatic qualitative improvement FeatUp achieves compared to several baselines. Our visualizations fit a 3-dimensional PCA on each image's low-resolution ViT features and use this PCA to map upsampled features into the same RGB space. We also show that this high-fidelity upsampling extends to higher PCA components in Figure C.5, and that FeatUp can improve small object retrieval in Figure C.8 in the Supplement.

Robustness across vision backbones Figure 5.6 demonstrates that FeatUp can upsample a variety of modern vision backbones. In particular, we show the implicit FeatUp features across a variety of backbones spanning transformers, convolutional nets, and both supervised and self-supervised models. Even though backbones like ResNet-50 do not precisely localize objects due to their large receptive fields, FeatUp can reasonably associate features to the correct object.

5.6.2 Transfer Learning for Semantic Segmentation and Depth Estimation

Next, we demonstrate that FeatUp can serve as a drop-in replacement for existing features in downstream applications. To demonstrate this, we adopt the widely used experimental procedure of using linear probe transfer learning to evaluate representation quality. More specifically, we train linear probes on top of low-resolution features for both semantic segmentation and depth estimation. We then freeze and apply these probes to upsampled

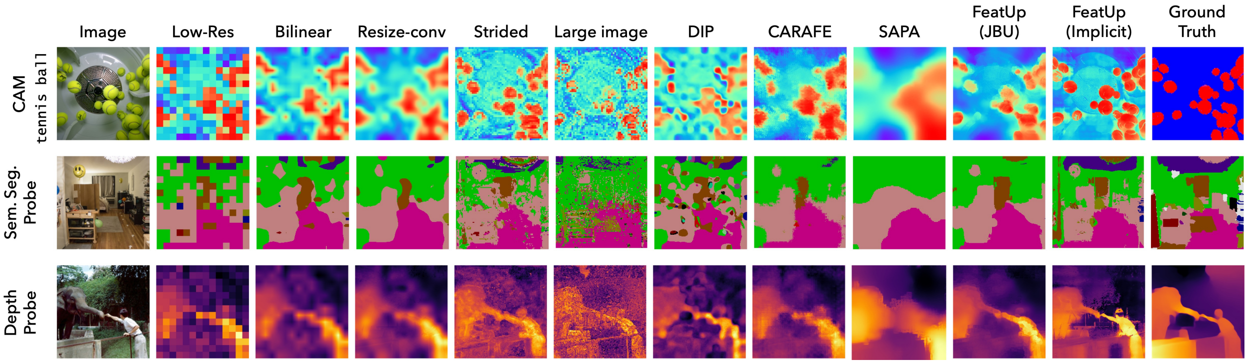


Figure 5.7: A comparison of different upsampling methods across each of the tasks considered in our analysis. FeatUp achieves significant improvements in resolution across each task.

features to measure performance improvement. If features are valid drop-in improvements, existing probes should work well without adaptation. For all experiments, we use a frozen pre-trained ViT-S/16 as the featurizer, upsample the features (14x14 → 224x224), and extract maps by applying a linear layer on the features.

For semantic segmentation, we follow the experimental setting of both [145, 216] and train a linear projection to predict the coarse classes of the COCO-Stuff (27 classes) training dataset using a cross-entropy loss. We report mIoU and accuracy on the validation set in Table 5.1. For depth prediction we train on pseudo-labels from the MiDaS (DPT-Hybrid) [217] depth estimation network using their scale- and shift-invariant MSE. We report root mean square error (RMSE) and the $\delta > 1.25$ metric which is common in monocular depth estimation literature. More specifically this metric is defined as the percentage of pixels with $\delta = \max(\frac{y}{y^*}, \frac{y^*}{y}) > 1.25$ where y is the depth prediction and y^* is the ground truth.

We stress that these linear probe evaluations show that FeatUp features can improve downstream tasks *without* re-training models. These analyses do not aim to create SOTA segmentation or depth networks. Both FeatUp variants outperform all baselines across all experiments, showing that either variant can be used as a drop-in replacement for existing features. Qualitatively, Figure 5.7 and Figures C.12 - C.13 in the supplement show cleaner, more cohesive predictions across both tasks.

5.6.3 Class Activation Map Quality

Attributing a model's predictions to specific pixels is crucial for diagnosing failures and understanding a model's behavior. Unfortunately, common interpretation methods like Class Activation Maps (CAM) are limited by the low res of the deep feature maps and cannot resolve small objects. We show that FeatUp features can be dropped into existing CAM analyses to yield stronger and more precise explanations. More specifically, we use the literature's established metrics, Average Drop (A.D.) and Average Increase (A.I.), that measure CAM quality (refer to Section C.11 in the Supplement for a detailed description of these metrics). Intuitively, A.D. and A.I. capture how much an image's most salient region changes the classification output. A good CAM should highlight regions with the greatest effect on the classifier's predictions, so censoring these regions will have the largest impact on the model's

Metric	Bilinear	Resize-conv	IndexNet	A2U	CARAFE	SAPA	FADE	FeatUp (JBU)
mIoU	39.7	41.1	41.5	41.5	42.4	41.6	<u>43.6</u>	**44.2**
mAcc	51.6	51.9	52.2	52.3	53.2	<u>55.3</u>	54.8	**55.8**
aAcc	78.7	79.8	<u>80.2</u>	79.9	80.1	79.8	**80.7**	**80.7**
Params (M)	13.7	+3.54	+12.6	+0.12	+0.78	+0.20	+0.29	+0.16
GFLOPs	16.0	+34.40	+30.90	+0.51	+1.66	+1.15	+2.95	+1.70

Table 5.2: Semantic segmentation results with the Segformer [218] architecture trained on the ADE20k train set and evaluated on the val set. FeatUp (JBU) outperforms the standard bilinear and resize-conv upsamplers in U-Net architectures, IndexNet [198], A2U [199], and other task-agnostic upsamplers (CARAFE [201], SAPA [202], FADE [200]). Additionally, our upsampler is competitive in parameter and floating-point operation count.

predictions (lower A.D., higher A.I.). Upsamplers are trained on the ImageNet training set for 2,000 steps, and we compute metrics across 2,000 random images from the validation set. We use a frozen pre-trained ViT-S/16 as the featurizer, and extract CAMs by applying a linear classifier after max-pooling. Upsampling is done (14x14 → 224x224) on the features themselves, and CAMs are obtained from these high-resolution maps. We report results in Table 5.1, and Figures 5.7, C.11.

5.6.4 End-to-end Semantic Segmentation

FeatUp not only improves the resolution of pre-trained features but can also improve models learned end-to-end. We adopt the experimental setting of [200, 202] to show that our JBU upsampler improves end-to-end performance on ADE20K semantic segmentation using the Segformer [218] architecture. Specifically, we train SegFormer on ADE20k [219, 220] (20,210 training and 2,000 val) for 160k steps. To validate that our setup matches that of existing literature despite numerical discrepancies, we also compute FLOPs for SegFormer with various upsamplers in Table 5.2. These counts are comparable with those in [221], confirming our architectural setup. We report mean IoU, mean class accuracy (mAcc), and all-pixel accuracy (aAcc) against several recent baselines in Table 5.2 including IndexNet [198], A2U [222], CARAFE [201], SAPA [202], and FADE [200] in addition to more standard bilinear and resize-conv operators. Figure C.14 in the Supplement shows examples of segmentation predictions across these methods. FeatUp consistently outperforms baselines with fewer added parameters, showing that FeatUp can also improve a broader, jointly trained architecture.

5.7 Chapter Conclusion

We present FeatUp, a novel approach to upsample deep features using multiview consistency. FeatUp solves a critical problem in computer vision: deep models learn high quality features but at prohibitively low spatial resolutions. Our JBU-based upsampler imposes strong spatial priors to accurately recover lost spatial information with a fast feedforward network based on a novel generalization of Joint Bilateral Upsampling. Our implicit FeatUp can learn high quality features at arbitrary resolutions. Both variants dramatically outperform a wide range of baselines across linear probe transfer learning, model interpretability, and end-to-end semantic segmentation.

Chapter 6

DenseAV: Unsupervised Visual Grounding of Sound and Language in Coss-Modal Representations

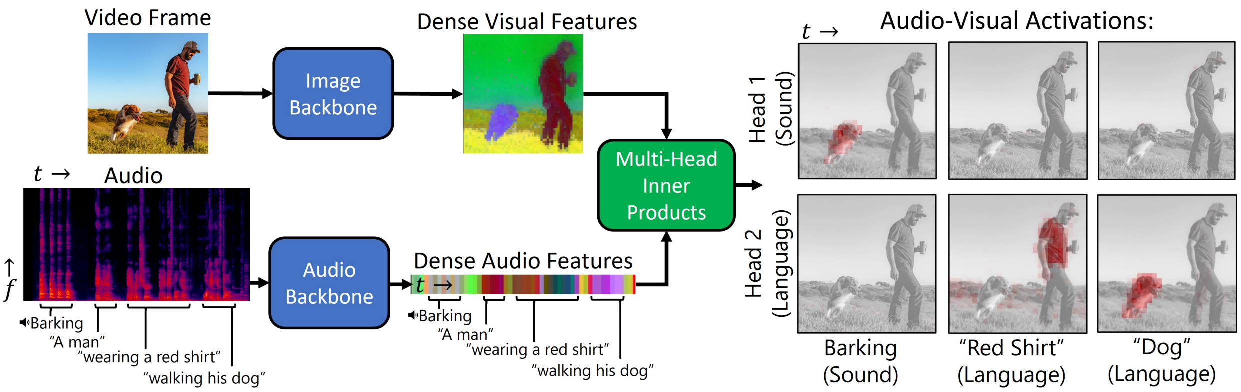


Figure 6.1: Visual overview of the DenseAV algorithm. Two modality-specific backbones featurize audio and visual signals. We introduce a novel generalization of multi-head attention to extract attention maps that discover and separate the "meaning" of spoken words and the sounds an object makes. DenseAV performs this localization and decomposition solely through observing paired stimuli such as videos.

6.1 Website and Video

For a quick video overview and blog post of this chapter, see https://mhamilton.net/denseav.html

6.2 Chapter Summary

In Chapters 3 and 4 we saw that deep representations have the power to connect similar objects together at very high spatial resolution (using Chapter 5) without humans in the loop. In this chapter, we take this idea one step further by connecting objects across two modalities: audio

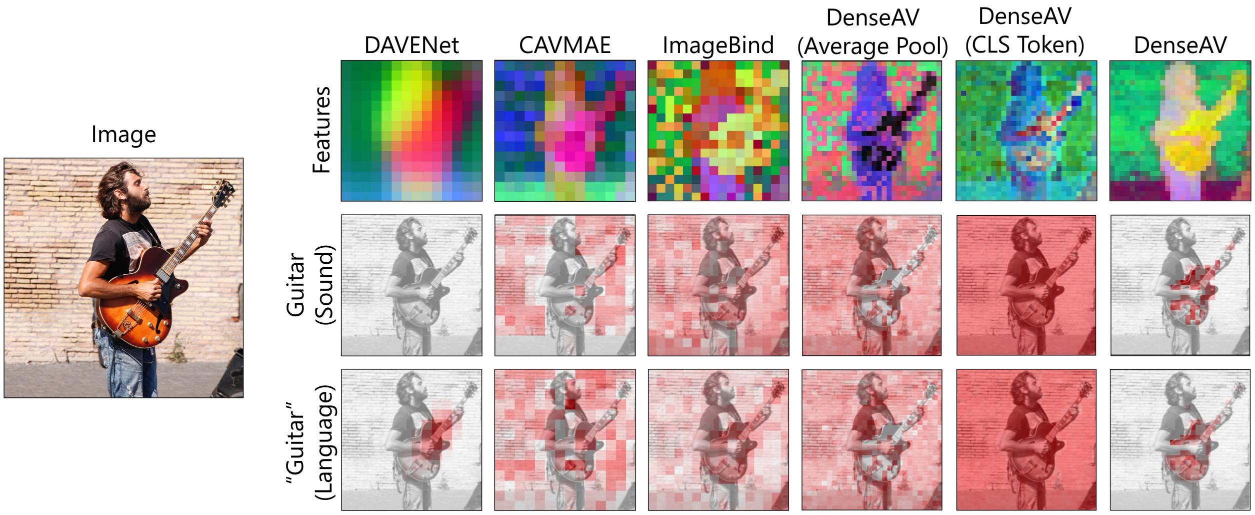


Figure 6.2: Qualitative comparison of several modern architectures for associating audio and video modalities. Only DenseAV learns a high-resolution and semantically aligned set of local features. This allows us to perform speech and sound prompted semantic segmentation using only the inner products between deep features. Other approaches, such as ImageBind, do not show aligned local feature maps. Approaches that do show some localization capabilities, like DAVENet, do not generalize to sound and language, and do not achieve the high-resolution localization capabilities of DenseAV. Dense features are visualized using PCA as described in Chapter 4

and video. In particular, we present DenseAV a novel dual encoder grounding architecture that learns high-resolution semantically meaningful and audio-visual aligned features solely through watching videos. We show that DenseAV can discover the "meaning" of words and the "location" of sounds without explicit localization supervision. Furthermore, it automatically discovers and distinguishes between these two types of associations without supervision. We show that DenseAV's localization abilities arise from a new multi-head feature aggregation operator that directly compares dense image and audio representations for contrastive learning. In contrast many other systems that learn "global" audio and video representations cannot localize words and sound. This concept is a multi-modal generalization of the key ideas introduced in Chapter 4. Finally, we contribute two new datasets to improve the evaluation of AV representations through speech and sound prompted semantic segmentation. On these and other datasets we show DenseAV dramatically outperforms the prior art on speech and sound prompted semantic segmentation. DenseAV outperforms the current state-of-the-art ImageBind on cross-modal retrieval using fewer than half of the parameters.

6.3 Introduction

In Chapters 3 and 4 we saw that deep representations have the power to connect similar objects together and that this was enough to discover visual objects from scratch. However, Chapter 4 could not automatically associate these found objects with language. Amazingly, the objects were almost identical to those discovered by humans just with arbitrary names like “Object 12”. In this chapter we show that discovering visual objects is just the start of whats possible by analyzing the relationships between deep representations, by generalizing this concept to multimodal relationships.

Associating audio and video events is a fundamental task in human perception. As infants develop, the synchronization and correspondence of visible sounds enables multi-modal association – a voice with a face, and a “moo" with a cow [223]. Later, as they acquire language, they associate spoken words with objects they represent [224, 225]. Amazingly, these association abilities, constituting speech recognition, sound event recognition, and visual object recognition, develop without much direct supervision. This work aims to create a model with this capability by learning high-resolution, semantically meaningful, audio-visually (AV) aligned representations. Features with these properties can be used to discover fine-grained correspondences between modalities without localization supervision or prior knowledge of the semantic representation of language, just like how the uni-modal relationships of Chapter 4 could discover visual objects.

Consider the spoken caption and accompanying sounds of the image shown in Figure 6.1. We wish to “ground” both the speech and the sounds by identifying them with the corresponding visual objects. For instance, both the spoken word “dog” and the sound of a bark in the audio signal should be associated with the pixels of the dog in the visual signal if present. We seek high quality local representations where this behavior, which is notably absent from popular approaches in the literature, emerges from simple inner products between cross-modal features.

To achieve this, we make three innovations. First, we introduce DenseAV, a dual-encoder architecture that computes a dense similarity volume over audio and visual features. Looking at a slice of this similarity volume for a spoken word, as in Figure 6.1, we can visualize the AV activation strength between a word or sound and an image’s pixels. The novelty we introduce is to extend this dense similarity mechanism to have multiple similarity volume heads, much like those of multi-head attention. This allows each head to specialize on a particular type of coupling between the visual and audio modalities. Interestingly, we discover that if we give DenseAV two heads and train on a dataset that contains both language and sound, the heads naturally learn to distinguish language from more general sound using only cross-modal supervision. For example, as shown in Figure 6.1, head 1 focuses on sounds, such as a dog bark, emitted by visible objects, whereas head 2 focuses on speech, such as the word “dog", that refers to visible objects.

Second, we show the importance of the “aggregation function” one uses to create a summary similarity score between an audio clip and a video frame for contrastive learning. The traditional choices, using inner products between global representations such as class tokens [112, 226, 227] or pooled features [228, 229], do not promote AV alignment of dense local features. Because of this, several popular audio-video backbones that excel on cross-modal retrieval *cannot* directly associate objects and sounds using their local features. This limits their ability to be used for downstream tasks such as semantic segmentation, sound localization, or unsupervised language learning and discovery.

Third, we introduce two semantic segmentation datasets to evaluate visual grounding with AV representations for speech and (non-speech) sounds. We build these datasets from the high-quality segmentation masks provided by the ADE20K dataset [230] and measure

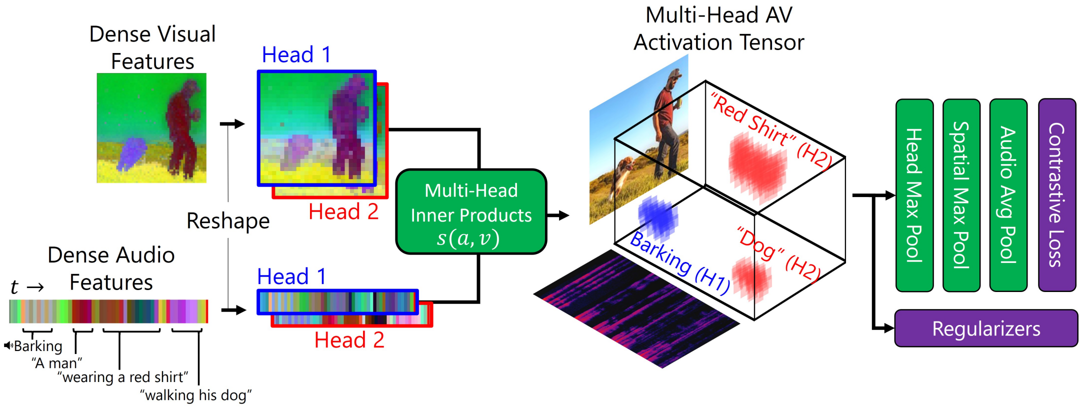


Figure 6.3: Architectural overview of our multi-head attention aggregator. Dense feature maps are split into K heads ($K = 1, 2$) in our experiments. We form an AV activation tensor by taking the inner-products of each head's features across the spatial and temporal extent of the visual and audio signals respectively as in Equation 6.1. We then aggregate this similarity volume into a single similarity score by max-pooling head and spatial dimensions and average-pooling audio dimensions. Our approach aims to encourage the network to identify specific shared objects between the audio and visual modalities. In particular, max-pooling of heads disentangles sound and language, and max-pooling spatial dimensions helps localize objects.

mean average precision (mAP) and mean intersection over union (mIoU) on a binary mask prediction task. This evaluation is simpler and more thorough than previous efforts to measure visual grounding such as the concept counting metrics of [231] and the "pointing games" of [232–234] that only check if a heatmap's peak occurs within a target box or segment. Furthermore, our evaluation avoids brittle word-net ontologies [235], clustering, Wu and Palmer distance [236], threshold choices, and a variety of other complicating factors.

To summarize, our main contributions are as follows:

- We introduce DenseAV, a novel self-supervised architecture that learns high-resolution AV correspondences.
- We introduce a local-feature-based image similarity function that significantly improves a network's zero-shot localization ability compared to common strategies such as average pooling or CLS tokens.
- We introduce new datasets for evaluating speech and sound prompted semantic segmentation. We show DenseAV significantly outperforms the current state-of-the-art on these tasks as well as on cross-modal retrieval.
- We discover that our multi-head architecture naturally disentangles audio-visual correspondence into sound and language components using only contrastive supervision.

6.4 Related Work

Audio-visual (AV), text-visual, and other multi-modal models have a long history [237, 238], and have recently surged in popularity [239]. Broadly speaking DenseAV is an audio-video contrastive learning architecture; this class of methods learns AV representations by aligning paired signals and pushing apart unpaired signals [240, 241]. Of the models in this class, several stand out for their ability to localize sounds [233, 242, 243] or capture the semantics of language [231, 244]. Many models in this class compare AV signals using inner products between "global" representations formed by pooled deep features [229, 245, 246], or class tokens [227, 243, 244, 247, 248]. Most notably, ImageBind has gained popularity due to its state-of-the-art performance on a variety of tasks and datasets and unified class-token-based contrastive architecture. In this work we show that many of these architectures do not show strong localization properties in their local features, despite excelling at cross-modal retrieval on a "global" level. This limits their applicability to new out-of-domain sounds, sounds that don't have a textual representation, and low-resource languages. We diverge from these works by directly supervising local tokens. In particular, we build on previous works [231, 233] that show max-pooling improves localization capabilities and introduce a new multi-head aggregation operator that generalizes previous losses using a self-attention-like operator [111].

Another class of methods discover structure in signals through uni- and multi-modal clustering. Early works on audio clustering [249] discovered meaningful utterances without supervision. Similar visual analyses, including those of Chapter 4 have discovered visual objects [250–253]. Recent works have applied these ideas to the AV domain [254, 255], but do not focus on extracting *high-resolution* AV representations.

Finally, several works investigate generative audio-video learning. The Sound of Pixels [256] generates the sound of a specific object using a source separation loss. Newer approaches using GANs [257, 258], and diffusion models [247, 259, 260] have generated audio from video and vice versa. Here we focus on improving the local representations of contrastive learners because of their relative scalability, simplicity, and ability to learn high-quality representations.

6.5 Methods

At a high level, DenseAV tries to determine when a given audio and visual signal belong "together" using dense audio-visual representations. To perform this task robustly, DenseAV must learn to predict the contents of an audio signal from a visual signal and vice versa. Doing so causes DenseAV to learn dense modality-specific features that capture the mutual information shared between the modalities [261]. Once learned, we can directly query these informative features to perform speech and sound prompted semantic segmentation as illustrated in Figure 6.1.

More specifically, DenseAV is built from two modality-specific deep featurizers. These backbones produce temporally varying audio features across an audio clip and spatially varying video features for a single randomly selected frame. Our loss computes a similarity between audio and visual signals based on the intuition that two signals are similar if they have a variety of strong couplings or shared objects. More formally, we form a scalar similarity for a pair of audio and video signals by carefully aggregating a volume of pairwise inner products between dense features. We use the InfoNCE [101] contrastive loss to encourage similarity between "positive" pairs of signals and dissimilarity between "negative" pairs formed by in-batch shuffling. Figure 6.3 graphically depicts this loss function and subsequent sections detail each component of our architecture.

6.5.1 Multi-Headed Aggregation of Similarities

DenseAV's key architectural distinction is its loss function that directly supervises the "local" tokens of the visual and audio featurizers. This is a significant departure from other works [226, 227, 247, 262–264] that pool modality specific information into "global" representations prior to the contrastive loss. Unlike prior works, our loss function aggregates the full pairwise similarities between the local tokens into an aggregate measure of similarity for a given pair of audio and visual signals. We show in Figure 6.2 that this architectural choice enables DenseAV's local features to align across modalities whereas other approaches such as average pooling, class tokens, and SimPool [265] do not.

We first describe our loss function informally and definite it more precisely in the next paragraph. Our loss function computes the (un-normalized) inner product between every pair of visual and audio features to form a "volume" of inner products. This volume represents how strongly each part of an audio signal "couples" to each part of a visual signal. We aim to find many large couplings between positive pairs of audio and visual signals. Ideally, these couplings should connect visual objects with their references in the audio signal. Conversely, we do not want to find couplings between negative pairs of signals. To compute a single global coupling strength for a pair of signals, we aggregate this volume of pairwise similarities into a single number. There are myriad ways to aggregate this volume ranging from "soft" average-pooling to "hard" max-pooling. Average pooling yields dense gradients and can improve convergence speed and stability. However, max-pooling allows the network to focus on the *best* couplings regardless the object's size or a sound's duration. Our aggregation function combines the benefits of average and max pooling by max-pooling visual dimensions and average pooling audio dimensions as proposed in [231]. Intuitively speaking, this averages the strongest image couplings over an audio signal. It allows small visual objects to have large effects yet provides a strong training gradient to many regions of the signals. Finally, we draw inspiration from multi-head self-attention [111] and generalize this operation to multiple "heads" that we max-pool before pooling the visual and audio dimensions. This allows DenseAV to discover multiple "ways" to associate objects across modalities.

More formally, let $\mathcal{S}(a, v) \in \mathbb{R}$ represent the similarity between a tensor of audio features $a \in \mathbb{R}^{CKFT}$ of size (**C**hannel × **K**-heads × **F**requency × **T**ime) and a tensor of visual features $v \in \mathbb{R}^{CKHW}$ of size (**C**hannel × **K**-heads × **H**eight × **W**idth). To define this scalar similarity score, we first create a local similarity volume, $s(a, v) \in \mathbb{R}^{kfthw}$. For simplicity, we consider the aggregated similarity between a single image and audio clip but note one can easily generalize this to max-pool over video-frames. We define the full pairwise volume of similarities as:

$$s(a,v) \in \mathbb{R}^{kfthw} = \sum_{c=1}^{C} a[c,k,f,t] \cdot v[c,k,h,w] \tag{6.1}$$

Where $a[c,k,f,t]$ represents the value of a at location $[c,k,f,t]$ and $\cdot$ is scalar multiplication. We aggregate this similarity volume into a single score $\mathcal{S}(a,v) \in \mathbb{R}$:

$$\mathcal{S}(a,v) = \frac{1}{FT} \sum_{f=1}^{F} \sum_{t=1}^{T} \max_{k,h,w} \left(s(a,v)[k,f,t,h,w] \right) \tag{6.2}$$

We note that this operation can be viewed as a multi-head generalization of the MISA loss of [231], and a multi-head multi-time generalization of the MIL loss of [233].

6.5.2 Loss

We can use the similarity between audio and visual signals defined in Equation 6.2 to construct a contrastive loss. We follow recent works [247, 266, 267] and use the temperature-weighted InfoNCE [101] to encourage similarity between positive pairs of signals and dissimilarity between negative pairs. In DenseAV, we form B positive pairs by splitting the audio and visual components of a **B**atch of training data. We form $B^2 - B$ negative pairs by comparing a signal to all of the other signals in the training batch. More formally let $(a_b, v_b)_1^B$ be B pairs of audio and visual signals. The visual-retrieval term of our InfoNCE loss is then:

$$\mathcal{L}_{A \to V} = \frac{1}{2B} \sum_{b=1}^{B} \left(\log \frac{\exp\left(\gamma \mathcal{S}(a_b, v_b)\right)}{\sum_{b'=1}^{B} \exp\left(\gamma \mathcal{S}(a_b, v_{b'})\right)} \right) \tag{6.3}$$

Where $\gamma \in \mathbb{R}^+$ is a trainable inverse temperature parameter. We symmetrize this loss by adding the analogous audio-retrieval term, $\mathcal{L}_{V \to A}$, which iterates over negative *audio* signals in the denominator.

6.5.3 Audio and Visual Featurizers

The core of DenseAV is two modality-specific backbone networks. We use the DINO vision transformer [226] with ImageNet pretrained weights (without labels) to provide a strong, yet fully unsupervised, vision backbone. Unlike other approaches that use CLIP [263] as a backbone, DINO does not require paired text captions and learns from unlabeled images only. Practically, we find that DINO outperforms CLIP because of its better-behaved local tokens [268], an effect we explore in the Supplement. We append an additional layer norm operation across the channel dimension [213] and a 1×1 Convolution to DINO. The layer-norm and 1×1 convolution ensure the architecture does not start with a saturated loss function. We use the HuBERT audio transformer [269] as DenseAV's audio backbone. HuBERT operates on waveforms and is trained on the LibriSpeech [270] dataset using only self-supervision. Hubert outputs a single feature per time frame, corresponding to $F = 1$ in Section 6.5. Though HuBERT was only trained on speech, its audio features can be fine-tuned for more general sounds, much like how vision backbones can be fine-tuned for new datasets [271]. As in the visual branch, we append a channel-wise LayerNorm block and two 3×3 convolutions to the audio branch. These layers help the network avoid saturation and speed convergence. Furthermore, the two convolutions help the model aggregate information, which reduces the cost of the pairwise feature comparison used in our loss function. We refer to these added layers after the pretrained backbones as the "aligners" in later sections.

6.5.4 Regularizers

Disentanglement Regularizer, $\mathcal{L}_{Dis}$: We add a small regularization term to encourage each head of Equation 6.1 to specialize and learn independent types of audio-visual associations. Interestingly we find that our 2-head model naturally learns to distinguish the meaning of words with one head and capture the sounds objects produce with another head. To further encourage this unsupervised discovery of concepts, we penalize the network when multiple attention heads are simultaneously active. More precisely, let $(a_b, v_b)_1^B$ be a **B**atch of B paired audio and visual signals. Our disentanglement loss for two heads is then:

$$\mathcal{L}_{Dis} = \text{Mean}(|s(a_b, v_b)[1] \circ s(a_b, v_b)[2]|) \tag{6.4}$$

Where $\circ$ is elementwise multiplication and $|\cdot|$ is the elementwise absolute value function. $[k]$ mirrors PyTorch slicing notation and refers to selecting the activations for only the kth attention head. Intuitively, this loss encourages one head to be silent if the other head is active and is a "cross-term" generalization of the l^2 regularizer [272] for encouraging activation shrinkage. When $K > 2$ we average contributions from every combination of heads. We ablate this, and our decision to max-pool heads in Table 6.3.

Stability Regularizers, $\mathcal{L}_{Stability}$: Finally, we add several other small regularization terms to encourage stable convergence. We detail and ablate these terms in the Supplement. Briefly, these terms include standard regularizers like Total Variation [215] smoothness over time and non-negative pressure to encourage the network to focus on similarity instead of dissimilarity. In addition, we add a regularizer to prevent the calibration temperature, γ, from drifting too quickly, and a regularizer to discourage activations during silence and noise. In the supplement we show that each regularizer alone does not have a dramatic effect on final metrics but together they can stop collapses during training.

Combining these losses into a single loss function yields:

$$\mathcal{L} = \mathcal{L}_{A\to V} + \mathcal{L}_{V\to A} + \lambda_{Dis}\mathcal{L}_{Dis} + \mathcal{L}_{Stability} \tag{6.5}$$

In our experiments we use $\lambda_{Dis} = 0.05$ and refer interested readers to the supplement for the details of our small stability regularizer, $\mathcal{L}_{Stability}$.

6.5.5 Training

In our experiments we train DenseAV and relevant baselines on the AudioSet [273] dataset for sound prompted segmentation and AudioSet retrieval. We train on the PlacesAudio [274] dataset for speech prompted segmentation, PlacesAudio retrieval, and the ablation studies of Table 6.4. In our disentanglement experiments of Table 6.3 and feature visualizations of Figures 6.1 and 6.2 we train on both AudioSet and PlacesAudio so that DenseAV can be familiar with both language, the prominent audio signal in PlacesAudio, and more general

Method	Speech Semseg.		Sound Semseg.	
	mAP	mIoU	mAP	mIoU
DAVENet [231]	32.2%	26.3%	16.8%	17.0%
CAVMAE [229]	27.2%	19.9%	26.0%	20.5%
ImageBind [247]	20.2%	19.7%	18.3%	18.1%
Ours	**48.7%**	**36.8%**	**32.7%**	**24.2%**

Table 6.1: **Speech and Sound prompted semantic segmentation**. We analyze the quality of local features using two prompted semantic segmentation tasks. We prompt networks with speech of the form "a picture of a(n) [Object]" to determine whether local feature inner products can segment objects in the ADE20K dataset by name. We create sound prompts for a given ADE20K class using a curated mapping from the ADE20K ontology to the VGGSound ontology. DenseAV's local features perform significantly better than all baselines investigated. We bold "first place" results and underline "second place" results.

sounds from AudioSet. In these experiments we sample training data from these two corpora, so each batch has an even split between AudioSet and PlacesAudio.

Warming up Aligners: We find that we can dramatically improve the stability by first training the added aligners (convolutions and layer norms) for 3000 steps while keeping pretrained DINO and HuBERT backbones fixed. This allows the aligners to adapt to these intelligent backbones before modifying each backbone's sensitive weights. We use random resize crops, color jitter, random flips, and random greyscaling as image augmentations. We randomly sample a single video frame to feed to our visual branch. Audio clips are converted to single-channel format and are trimmed or padded with silence to create uniform 10 second clips. We re-sample audio clips according to the requirements of the backbone models used. For HuBERT, we re-sample to 16KhZ. We train on 8 V100 GPUs with an effective batch size of 80, and aggregate negative samples on all GPUs prior to computing the loss to ensure efficient parallelization. We provide additional training information and hyperparameters in the supplement.

Full Training: After warming up the aligners, we train the full model for an additional 800,000 steps using the same loss, batch-size, and training logic. We train all aligner weights and fine-tune all HuBERT audio backbone weights. We use low rank adaptation (LoRA) [275] to fine-tune the "Q", "K", and "V" layers of the DINO visual backbone attention blocks. This allows us to efficiently adapt DINO and stabilize the training as it is quite easy to collapse the carefully trained DINO weights. We use a LoRA rank of 8.

6.6 Experiments

To evaluate AV representation quality, we perform a variety of analyses including comparative activation visualization, quantitative measurements of speech and sound prompted semantic segmentation, and cross-modal retrieval. Additionally, we quantify our observation that DenseAV can distinguish the meanings of words (language), from the sounds of objects (sound) without supervision.

To adequately measure a representation's AV alignment quality, we found it necessary to introduce two evaluation datasets that measure speech and sound prompted semantic segmentation performance. Our two datasets introduce pairs of speech and sound prompts coupled with matching images and segmentation masks derived from ADE20K. We create these datasets because previous works [231] have not published their datasets or evaluation code. However, we use an experimental setting from the literature for our cross-modal retrieval experiments.

We compare against a variety of prior art including the popular state-of-the art multi-modal retrieval network, ImageBind [247]. We also compare against CAVMAE [229], a leading multimodal backbone trained specifically for AudioSet retrieval, and DAVENet [231], which is trained to localize the meanings of words. We include two other baselines [254, 274] which have reported cross modal retrieval metrics on Places Audio. Finally, we compare our multi-head aggregation strategy to common "global" retrieval methods such as inner products between class-tokens, average-pooled tokens, and SimPooled[265] tokens. We note that SimPool achieves state-of-the-art localization results when compared to 14 other pooling methods. Nevertheless, our multi-head aligner yields better localization results than any of these "global" methods.

6.6.1 Qualitative Comparison of Feature Maps

Our first experiment in Figure 6.2 highlights the dramatic differences in quality between DenseAV's features and other approaches in the literature. DenseAV is the only backbone whose local tokens are semantically meaningful and show cross-modal alignment for speech and sound. Though both CAVMAE and ImageBind show high-quality retrieval performance, neither shows high quality aligned local tokens. As a result, DenseAV can associate and localize both sound and language significantly better than other backbones. DAVENet shows coarse correspondences between language and visual objects but cannot associate sound with visual objects and does not match DenseAV's high resolution maps. Furthermore, the right half of Figure 6.1 demonstrates that DenseAV naturally discovers and separates word semantics from the sound of objects without labels to supervise this separation. In the supplement, we provide additional visualizations of all backbones considered across a wide range of words and sounds.

6.6.2 Speech Prompted Image Segmentation

Dataset: We introduce a speech prompted segmentation dataset using the ADE20K dataset, which is known for its comprehensive ontology and pixel-precise annotations [230]. From this dataset, we curate an evaluation subset of image-class pairs by sampling up to 10 images for each object class in ADE20K, excluding images where the selected class was tiny ($<5\%$ of pixels). We only consider classes with at least 2 images that pass the tiny object criterion. For each class and image, we formed a binary target mask by selecting the semantic segmentation mask for that class. This resulted in 3030 image-object pairs spanning 478 ADE20K classes.

We created paired speech signals by speaking the prompt "A picture of a(n) [object]" where [object] is the name of the ADE20K class. We create clear, controlled, and consistent audio prompts using Microsoft's neural text to speech service [276]. This service also provides

Method	Places Acc. @10		AudioSet Acc. @10	
	I → A	A → I	I → A	A → I
[274]*	46.3%	54.8%	-	-
[254]*	54.2%	56.4%	-	-
DAVENet [231]*	52.8%	60.4%	-	-
CAVMAE [229]	81.7%	77.7%	55.7%	50.7%
ImageBind [247]	1.10%	1.10%	64.5%	66.5%
Ours	**94.2%**	**94.3%**	**69.8%**	**68.1%**

Table 6.2: **Cross-modal retrieval using 1000 evaluation videos from the PlacesAudio and AudioSet validation datasets**. DenseAV dramatically outperforms all approaches tested in all metrics. Most notably, the state-of-the-art image retrieval foundation model, ImageBind, is incapable of recognizing speech. We note that the ImageBind authors do not publish retraining code, so we evaluate their largest pretrained model. Models with a * indicate that they have been previously reported in the literature. Other numbers are calculated by using pretrained models when available or from training with the author's official training scripts.

exact timing of the "[object]" utterance within the broader prompt and ensures each class is measured equally. Grammar was manually verified for the utterances to ensure proper singular/plural and a/an agreement with the class name. We release images, masks, and audio prompts for reproducibility.

Evaluation Measure: We evaluate methods based on how well their speech-prompted activations align with ground truth masks for the visual object's class. We quantify this with the binary Average Precision (AP) and Intersection over Union (IoU) metrics. These quantify how close activations match with the binary label mask from the ADE20K dataset. To compute an aggregate score over all of the object classes considered, we compute the mean average precision (mAP) and mean intersection over union (mIoU) by averaging AP scores across all object categories considered.

The mAP is particularly well suited for evaluating feature similarities because it is unaffected by monotonic transformations of the similarity scores. This eliminates the need for arbitrary thresholding and calibration. This is particularly important because many networks' inner products are not centered at zero, and the best thresholding strategy can be nontrivial, and dependent on the network and object class. Average Precision avoids these confounding factors and ensures a fair comparison across methods. Unfortunately, unlike the mAP, the mIoU metric requires selecting a threshold. To ensure our mIoU measurement is similarly invariant to monotonic transformations we evaluate 20 uniformly spaced thresholds between the smallest and largest activations of each model. For each baseline, we report results for the best threshold to ensure a fair comparison between all networks considered.

Implementation: We compute image heatmaps by evaluating each modality-specific network on the image-audio pairs from our dataset. We extract dense features from the final layer of each network and form their similarity volume according to Equation 6.1. For DenseAV we max-pool the head dimension to properly compare with single-headed models. We average activations over the temporal extent of the "[object]" utterance using the word timing

Method	Pred. Dis.	Act. Dis.
No $\mathcal{L}_{Dis}$, No Head Max Pool	64.1%	70.3%
No $\mathcal{L}_{Dis}$	**99.9%**	<u>86.5</u>%
Ours	**99.9%**	**91.2%**

Table 6.3: Quantitative ablation study of the impact of max-pooling attention heads and adding our disentanglement loss, $\mathcal{L}_{Dis}$. Intuitively, max-pooling attention heads allows each head to specialize on its own specific set of triggers. Our disentanglement loss further encourages the heads to operate independently and orthogonally.

information from the ground truth audio clip. This creates a heatmap over the image features that can be bi-linearly resized to the original image's size. We then compare these per-pixel activation scores to ground truth object masks from our dataset.

Results: In Speech mAP and mIoU columns of Table 6.1 we show that DenseAV achieves a **51% (+16.5 mAP)** relative increase in speech-prompted semantic segmentation over previous methods. Approaches that use global token based contrastive strategies such as CAVMAE and ImageBind perform particularly poorly in this task, and this observation aligns with the qualitative results of Figure 6.2.

6.6.3 Sound Prompted Image Segmentation

Dataset: To evaluate how well deep features localize sound, we build on Section 6.6.2 and create a dataset of sound prompts that align with ADE20K classes. We first select the same (large) image-object pairs from ADE20K. We then create a mapping between the ADE20K and VGGSound [277] ontologies. To compute a robust mapping, we first embed ADE20K class names and VGGSound class names with the GPT Ada 2 text embedding model [278]. For each ADE20K class, we create a list of at most three candidates from the VGGSound ontology that have a cosine similarity ($> .85$). We then manually review these candidates to select the best VGGSound class for each ADE20K class and remove any spurious or mistaken matches. This produces a set of 95 ADE20K classes with strong matches in the VGGSound ontology. For each of our original 3030 image-object pairs we select a random VGGSound validation clip with a matching class according to our mapped ontology. This yields 106 image-object pairs across 20 ADE20K classes.

Evaluation Measure: We use the same mAP and mIoU evaluation metrics as Section 6.6.2, but instead average over the 20 ADE20K classes considered.

Implementation: We compute sound prompted image activations as in section 6.6.2 but with one key change: we average activations over the entire clip because we do not have ground-truth sound timing information.

Results: The "Sound mAP and mIoU" columns of Table 6.1 show that DenseAV achieves a **25% (+6.4mAP)** relative improvement in sound prompted segmentation compared to the prior art. Most notably, ImageBind's features cannot localize sound despite their high cross-modal retrieval performance learned from millions of hours of sound.

Method	Speech mAP	Places Acc. @10	
		V → A	A → V
Average Pool	20.1%	92.0%	91.2%
CLS Token	20.6%	86.4%	89.8%
SimPool [265]	35.3%	92.6%	92.8%
Multi-Head (Ours)	**48.2%**	**93.5%**	**93.8%**

Table 6.4: Quantitative ablation of different feature aggregation strategies. Though the common practice of average pooling and using a learned CLS token to aggregate features have little effect on retrieval performance, they dramatically degrade performance on speech prompted semantic segmentation.

6.6.4 Cross-Modal Retrieval

We show that DenseAV's representations are not only better for localization, but significantly outperform other approaches on cross-modal retrieval. We adopt the evaluation setting of [231] and measure cross modal retrieval accuracy at 1, 5, and 10 in a thousand-way retrieval task. In particular, we use the same thousand images from the validation set of [231] and also replicate this analysis on one-thousand random clips from the AudioSet validation data. Table 6.2 shows results for 1000-way retrieval tasks on both the Places Audio and AudioSet datasets. We show cross-modal accuracy at 10, but also show larger tables in the supplement that echo these results using accuracy at 1 and 5. DenseAV significantly outperforms all baselines across all metrics. Interestingly, DenseAV outperforms ImageBind with *less than half* of the trainable parameters and no reliance on text.

6.6.5 Measuring Disentanglement

We observe that DenseAV's heads naturally learn to differentiate audio-visual couplings that capture the meaning of words (language) and those that capture the sounds of objects (sound). Furthermore this effect generalizes to novel clips, including those with both sound and language as shown in Figure 6.1. We quantify this observation in two ways, the first measures if a head's average activation strength predicts whether a clip contains mainly "language" or "sound". The second method quantifies how often the "sound" head is incorrectly active when the "language" head should be active and vice versa. We leverage the fact that AudioSet dataset contains mostly clips with ambient sound and rarely contains language. In contrast, Places Audio is entirely language-based without external ambient sound. We note that these analyses are specifically for our architecture with two heads $K = 2$ and trained on both AudioSet and PlacesAudio data.

For both measures of disentanglement, we first compute a clip's aggregated similarity for each head. In particular, we remove the max-pooling over heads in Equation 6.2 to create a single-head similarity, $\mathcal{S}(a, v)_k$. We then min-max scale the scores of each head across both datasets to lie in the $[0, 1]$ interval, which we refer to as $\hat{\mathcal{S}}(a, v)_k$. Using these normalized scores, we can create metrics that capture how well a given head responds only to a specific dataset.

Our first metric measures how well a head's scores predict whether a clip is from the "sound" or "language" dataset. Let $(a_b, v_b)_1^B$ be tuples of paired audio and visual signals. let $l[k']_b$ be an indicator variable of whether the signal (a_b, v_b) arises from the sound dataset, AudioSet, $(k' = 1)$, or the language dataset Places Audio $(k' = 2)$.

$$\delta_{pred}(k, k') = \text{AP}\left((\hat{\mathcal{S}}(a_b, v_b)_k)_1^B, (l[k']_b)_1^B \right) \tag{6.6}$$

Where $AP(\cdot, \cdot)$ is the binary average precision with prediction and label arguments respectively. Intuitively, this measures whether the scores of head k are direct predictors of whether the data is from dataset k'. We can find the best assignment between heads and datasets such that each head is maximally predictive of the given dataset:

$$\text{PredDis} = \frac{1}{2} \max \left(\delta_{pred}(0, 0) + \delta_{pred}(1, 1),\ \ \delta_{pred}(1, 0) + \delta_{pred}(0, 1) \right) \tag{6.7}$$

The prediction disentanglement score, PredDis, is a percentage that ranges from 50% for completely entangled signals to 100% if one can perfectly classify the signals using the scores of either head. The maximum over the two possible assignments makes this metric invariant to permutations of the heads. We note that this metric, just like that of Chapter 4, is a Hungarian matching assignment [279] over two entries, a common technique to asses unsupervised classification performance [250, 253].

Our second measure quantifies "spurious activations" in the non-dominant head. A truly disentangled system should have a head that only fires on sound, and another head that only fires on language. We create another disentanglement measure, ActDis, by replacing δ_{pred} in Equation 6.7 with:

$$\delta_{act}(k, k') = 1 - \frac{1}{\sum_{b'} l[k']_{b'}} \sum_{b=1}^{B} \hat{\mathcal{S}}(a_b, v_b)_k \cdot l[k']_b \tag{6.8}$$

Intuitively, this measures the "inactivity" of head k on dataset k'. If head k is totally silent on dataset k' then $\delta_{act}(k, k') = 1$. Like PredDis, ActDis is a percentage ranging from 50% to 100% with 100% representing perfect disentanglement where the sound head is completely silent during the language clips, and vice versa.

Table 6.3 shows that DenseAV achieves near perfect predictive (99%) and activation (91%) disentanglement. It also shows that our disentanglement regularizer and max-pooling over heads improves DenseAV's natural ability to distinguish sound from language without supervision.

6.7 Chapter Conclusion

We presented DenseAV, a novel contrastive learning architecture that can discover the meaning of words and localize the sounds of objects using only video supervision. We are the first to observe both qualitatively and quantitatively that it's possible to disentangle the meaning of words from the sound of objects with only a contrastive learning signal. DenseAV's success stems from its novel multi-head attention aggregation mechanism that encourages its modality-specific backbones to create high-resolution, semantically meaningful, and AV aligned representations. These properties of DenseAV's representation are not seen in other state-of-the-art models in the literature. Consequently, DenseAV significantly surpasses other leading models in dense prediction tasks such as speech and sound-prompted semantic segmentation as well as in cross-modal retrieval.

Chapter 7

An Axiomatic Theory Connecting Model Explainability, Game Theory, and Feature Relationships

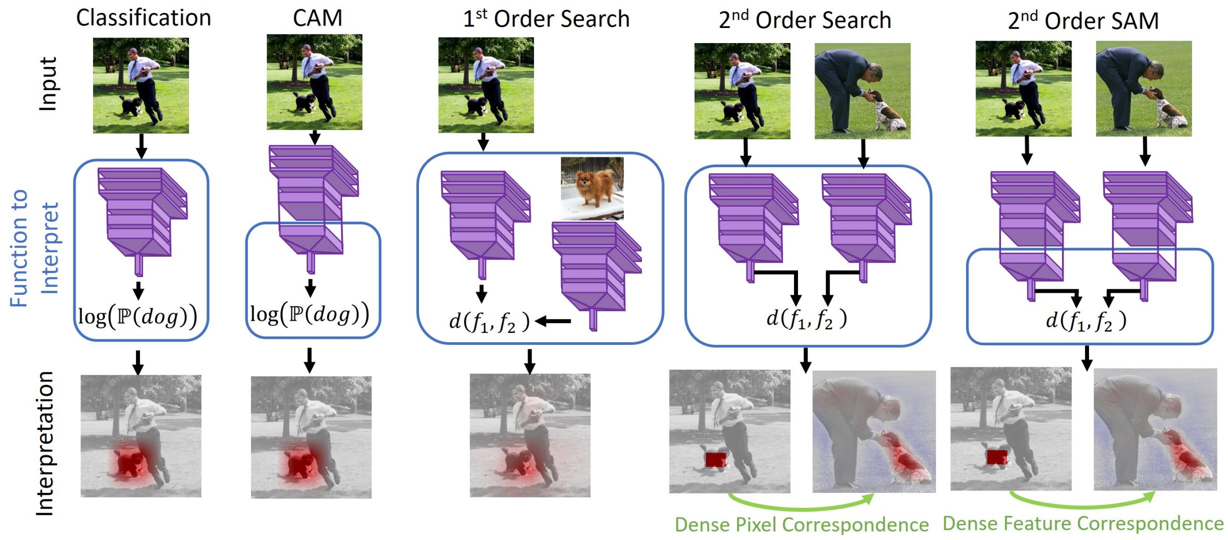


Figure 7.1: Architectures for search engine interpretability. Like classifier explanations, First-order search explanations yield heatmaps of important pixels for similarity (bottom row third column). Second order search interpretation methods yield a dense correspondence between image locations (last two columns). CAM (second column) is a particular case of Shapley value approximation, and we generalize it to yield dense correspondences (last column).

7.1 Website and Video

For a quick video overview and blog post of this chapter, see https://mhamilton.net/axiomatic.html

7.2 Chapter Summary

Visual search, recommendation, and contrastive similarity learning power technologies that impact billions of users worldwide. Chapters 3, 4, and 6 all show that inner-products between deep features derived from these systems have the unexpected ability to connect related objects across huge gaps, in time, artistic media, image context, and modality respectively. But the question remains: Why do these objects have this ability to visualize these connections at high spatial resolution when they are not trained with object localization supervision? We answer this question by introducing a theory connecting model explainability, cooperative game theory, and deep feature relationships. In short, this chapter shows that the feature correspondences at the heart of Chapters 3, 4, and 6, are precisely the unique way to "distribute the credit" for the prediction across the features.

More precisely, we show that the theory of fair credit assignment provides a *unique* axiomatic solution that generalizes several existing recommendation- and metric-explainability techniques in the literature. Modern model architectures can be complex and difficult to interpret, and there are several competing techniques one can use to explain a search engine's behavior. Using this formalism, we show when existing approaches violate "fairness" and derive methods that sidestep these shortcomings and naturally handle counterfactual information. More specifically, we show existing approaches implicitly approximate second-order Shapley-Taylor indices and extend CAM, GradCAM, LIME, SHAP, SBSM, and other methods to search engines. These extensions can extract pairwise correspondences between images from trained *opaque-box* models. We also introduce a fast kernel-based method for estimating Shapley-Taylor indices that require orders of magnitude fewer function evaluations to converge. Finally, we show that these game-theoretic measures yield more consistent explanations for image similarity architectures.

7.3 Introduction

Search, recommendation, retrieval, and contrastive similarity learning powers many of today's machine learning systems. These systems help us organize information at scales that no human could match. Furthermore, chapters 4, and 6 provide examples of how dense representations from these systems seem to capture detailed local relationships between objects even though no human gave them localization supervision. This chapter seeks to explain this mystery. In solving this, this chapter discovers a rich theoretical framework linking these inner products between representations with cooperative game theory and higher-order explanations of model behavior.

This theory not only explains the success of methods from previous chapters, but provides new insights to improve how we understand what search engines and recommendation systems learn. This is especially important given the recent surge in million and billion parameter contrastive learning architectures for vision and language, which underscore the growing need to understand these classes of systems [100, 280–283]. Like classifiers and regressors, contrastive systems face a key challenge: richer models can improve performance but hinder interpretability. In high-risk domains like medicine, incorrect search results can have serious consequences. In other domains, search engine bias can disproportionately and systematically hide certain voices [284–286].

Currently, there are several competing techniques to understand a similarity model's predictions [287–291]. However, there is no agreed "best" method and no a formal theory describing an "optimal" search explanation method. We show that the theory of fair credit assignment provides a uniquely determined and axiomatically grounded approach for "explaining" a trained model's similarity judgements. In many cases, existing approaches are special cases of this formalism. This observation allows us to design variants of these methods that better satisfy the axioms of fair credit assignment and can handle counterfactual or relative explanations. Though we explore this topic through the lens of visual search, we note that these techniques could also apply to text, tabular, or audio search systems.

This work identifies two distinct classes of search engine explainability methods. "First order" approaches highlight the most important pixels that contribute to the similarity of objects and "Second order" explanations provide a full correspondence between the parts of query and retrieved image. We relate first order interpretations to existing theory on classifier explainability through a generic function transformation, as shown in the third column of Figure 7.1. We find that second order explanations correspond to a uniquely specified generalization of the Shapley values [292] and is equivalent to projecting Harsanyi Dividends onto low-order subsets [293]. We use this formalism to create new second-order generalizations of Class Activation Maps [294], GradCAM [290], LIME [295], and SHAP [296]. Our contributions generalize several existing methods, illustrate a rich mathematical structure connecting model explainability and cooperative game theory, and allow practitioners to understand search engines with greater nuance and detail. We include a short video detailing the work at https://aka.ms/axiomatic-video. In summary we:

- Present the first uniquely specified axiomatic framework for *model-agnostic* search, retrieval, and metric learning interpretability using the theory of Harsanyi dividends.
- Show that our framework generalizes several existing model explanation methods [287, 290, 294, 295] to yield dense pairwise correspondences between images and handle counterfactual information.
- Introduce a new kernel-based approximator for Shapley-Taylor indices that requires about 10× fewer function evaluations.
- Show that our axiomatic approaches provide more faithful explanations of image similarity on the PascalVOC and MSCoCo datasets.

7.4 Background

This work focuses on search, retrieval, metric learning, and recommendation architectures. Often, these systems use similarity between objects or learned features [21] to rank, retrieve, or suggest content [15, 240, 282, 297]. More formally, we refer to systems that use a distance, relevance, or similarity function of the form: $d : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to quantify the relationship between items from sets $\mathcal{X}$ and $\mathcal{Y}$. In search and retrieval, $\mathcal{X}$ represents the space of search queries and $\mathcal{Y}$ represents the space of results, the function d assigns a relevance to each query

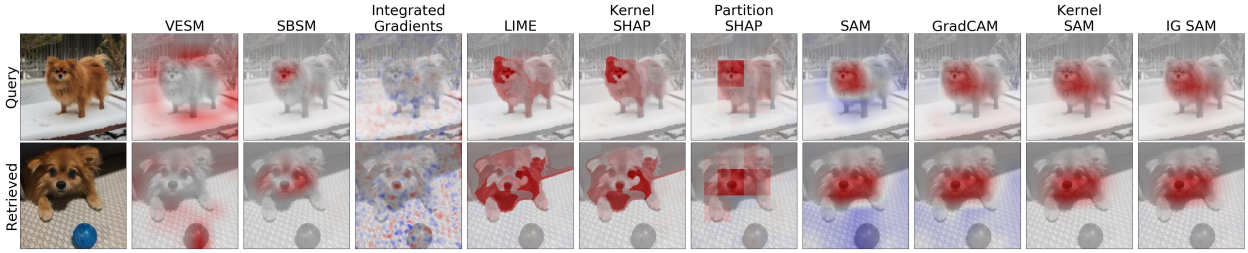


Figure 7.2: Comparison of first-order search interpretation methods which highlight pixels that contribute to similarity in red. Integrated Gradients (on pixels) struggles because well trained classifiers are invariant to minor pixel changes and have uninformative gradients.

result pair. Without loss of generality, we consider d as a "distance-like" function where smaller values indicate more relevance. The expression $\text{argmin}_{y\in\mathcal{Y}}\, d(x,y)$ yields the most relevant result for a query $x \in \mathcal{X}$.

Specializing this notion yields a variety of different kinds of ML systems. If $\mathcal{X} = \mathcal{Y} = \text{Range}(\mathcal{N}(\cdot))$ where $\mathcal{N}$ is an image featurization network such as ResNet50 [121], the formalism yields a visual search engine or "reverse image search". Though this work focuses on visual search, we note that if $\mathcal{X}$ is the space of character sequences and $\mathcal{Y}$ is the space of webpages, this represents web search. In recommendation problems, $\mathcal{X}$ are users and $\mathcal{Y}$ are items, such as songs or news articles. In this work we aim to extract meaningful "interpretations" or "explanations" of the function d.

7.4.1 Model Interpretability

The Bias-Variance trade-off [298] affects all machine learning systems and governs the relationship between a model's expressiveness and generalization ability. In data-rich scenarios, a model's bias dominates generalization error and increasing the size of the model class can improve performance. However, increasing model complexity can degrade model interpretability because added parameters can lose their connection to physically meaningful quantities. This affects not only classification and regression systems, but search and recommendation architectures as well. For example, the Netflix-prize-winning "BellKor" algorithm [299], boosts and ensembles several different methods making it difficult to interpret through model parameter inspection alone.

To tackle these challenges, some works introduce model classes that are naturally interpretable [300, 301]. Alternatively, other works propose *model-agnostic* methods to explain the predictions of classifiers and regressors. Many of these approaches explain the local structure around a specific prediction. [296] show that the Shapley value [302], a measure of fair credit assignment, provides a unique and axiomatically characterized solution to classifier interpretability (SHAP). Furthermore, they show that Shapley values generalize LIME, DeepLIFT [303], Layer-Wise Relevance Propagation [304], and several other methods [305–308]. Many works in computer vision use an alternative approach called Class Activation Maps (CAMs). CAM projects the predicted class of a deep global average pooled (GAP) convolutional network onto the feature space to create a low resolution heatmap of class-specific network attention. GradCAM [290] generalizes CAM to architectures other than GAP and can explain a prediction using only a single network evaluation. In Section 7.6 we show that CAM, GradCAM, and their analogue for search engine interpretability, [287], are also unified by the Shapley value and its second order generalization, the Shapley-Taylor index.

7.4.2 Fair Credit Assignment and the Shapley Value

Shapley values provide a principled and axiomatic framework for classifier interpretation. We briefly overview Shapley values and point readers to [309] for more detail. Shapley values originated in cooperative game theory as the **only** fair way to allocate the profit of a company to its employees based on their contributions. To formalize this notion we define a "coalition game" as a set N of $|N|$ players and a "value" function $v : 2^N \to \mathbb{R}$. In cooperative game theory, this function v represents the expected payout earned by cooperating coalition of players. [302] show that the unique, fair credit assignment to each player, $\phi_v(i \in N)$, can be calculated as:

$$\phi_v(i) := \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S)) \tag{7.1}$$

Informally, this equation measures the average increase in value that a player i brings to a coalition S by weighting each increase, $v(S \cup \{i\}) - v(S)$, by the number of ways this event could have happened during the formation of the "grand coalition" N. We note that this assignment, ϕ_v, is the *unique* assignment that satisfies four reasonable properties: **symmetry** under player re-labeling, no credit assignment to **dummy** players, **linearity** (or it's alternative **monotonicity**), and **efficiency** which states that Shapley values should sum to $v(N) - v(\emptyset)$ [310]. Intuitively, these axioms require that a fair explanation should treat every feature equally (Symmetry), should not assign importance to features that are not used (Dummy), should behave linearly when the value function is transformed (Linear), and should sum to the function's value (Efficiency).

Shapley Values provide a principled way to explain the predictions of a machine learning model. To connect this work to model interpretability, we can identify the "features" used in a model as the "players" and interpret the value function, $v(S)$, as the expected prediction of the model when features $N \setminus S$ are replaced by values from a "background" distribution. This background distribution allows for "counterfactual" or relative explanations [311].

7.5 Related Work

There is a considerable body of literature on model interpretability and we mention just a handful of the works that are particularly related. One of our baseline methods, [289], was one of the first to present a generic visual search engine explanation reminiscent of a Parzen-Window based estimator. [234] introduce a method for explaining classifiers based on meaningful perturbation and [312] introduce a method for improving interpretation for transformer-based classifiers. [287] lifted CAM to search engines and we find that our Shapley-Taylor based method aligns with their approach for GAP architectures. [313] and [314] use LIME and DeepSHAP to provide first-order interpretations of text but do not apply their methods to images. [315] introduce a distribution propagation approach for improving

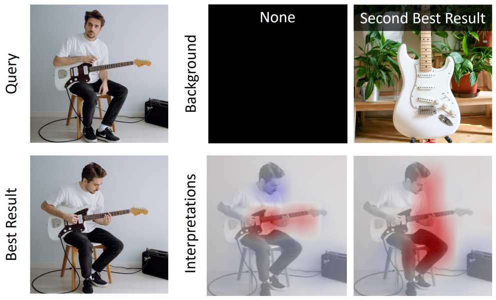


Figure 7.3: Explanations relative to a background distribution show why a result is better than an alternative. When asked why the best result (lower left) was better than the second best result (top right) our method correctly selects the player.

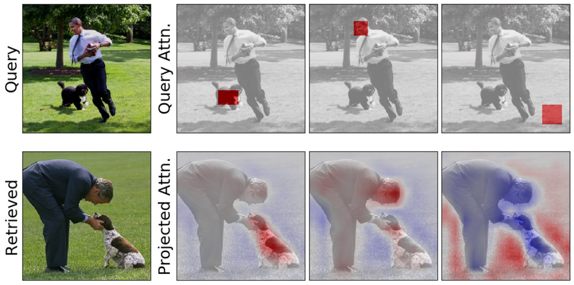


Figure 7.4: Visualization of how regions of two similar images "correspond" according to the second-order search interpretability method SAM. We can use this correspondence to transfer labels or attention between similar images.

the estimation of Shapley Values for deep models and can be combined with our approach. Many works implicitly use components that align with Shapley-Taylor indices for particular functions. Works such as [316–320] use feature correlation layers to estimate and utilize correspondences between images. We show these layers are equivalent to Shapley-Taylor indices on the GAP architecture, and this allows create a correlation layer that handles counterfactual backgrounds. Other recent works have used learned co-attention within transformer architectures to help pool and share information across multiple domain types [321]. [322] attempt to learn a variant of GradCAM that better aligns with axioms similar to Shapley Values by adding efficiency regularizers. The method is not guaranteed to satisfy the axioms but is more "efficient".

We rely on several works to extend Shapley values to more complex interactions. [293] generalized the Shapely value by introducing a "dividend" that, when split and distributed among players, yields the Shapley values. [323] introduces an equivalent way to extend Shapley values using a multi-linear extension of the game's characteristic function. [292] introduce the Shapley-Taylor index and show is equivalent to the Lagrangian remainder of Owen's multi-linear extension. Integrated Hessians [324] enable estimation of a second-order variant of the Aumann-Shapley values and we use this approach to create a more principled second-order interpretation method for differentiable search engines.

7.6 Unifying First-Order Search Interpretation Techniques

Though there is a considerable body of work on opaque-box classifier interpretability, opaque-box search engine interpretability has only recently been investigated [287, 288, 313]. We introduce an approach to transform opaque and grey-box classification explainers into search engine explainers, allowing us to build on the rich body of existing work for classifiers. More formally, given a similarity function $d : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and elements $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ we can find the "parts" of y that most contribute to the similarity by computing the Shapley values for the following value function:

$$v_1(S) : 2^N \to \mathbb{R} := d(x, mask(y, S)) \tag{7.2}$$

Where the function $mask(\cdot,\cdot) : \mathcal{Y} \times 2^N \to \mathcal{Y}$, replaces "parts" of y indexed by S with components from a background distribution. Depending on the method, "parts" could refer to image superpixels, small crops, or locations in a deep feature map. This formula allows us to lift many existing approaches to search engine interpretability. For example, let $\mathcal{X}$, and $\mathcal{Y}$ represent the space of pixel representations of images. Let the grand coalition, N, index a collection of superpixels from the retrieved image y. Let $mask(y, S)$ act on an image y by replacing the S superpixels with background signal. With these choices, the formalism provides a search-engine specific version of ImageLIME and KernelSHAP. Here, Shapley values for each $i \in S$ measure the impact of the corresponding superpixel on the similarity function. If we replace superpixels with hierarchical squares of pixels we arrive at Partition SHAP [325]. We can also switch the order of the arguments to get an approach for explaining the query image's impact on the similarity. In Figure 7.2 we qualitatively compare how methods derived from our approach compare to two existing approaches: SBSM [289] and VESM [288], on a pair of images and a MocoV2 based image similarity model. In addition to generalizing LIME and SHAP we note that this approach generalizes VEDML [287], a metric-learning adaptation of CAM:

Proposition 7.6.1. *Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^{CHW}$ and represent the space of deep network features where C, H, W represent a channel, height, and width of the feature maps respectively. Let the function $d := \sum_c GAP(x)_c GAP(y)_c$. Let the grand coalition, $N = [0, H] \times [0, W]$, index the spatial coordinates of the image feature map y. Let the function $mask(y, S)$ act on a feature map y by replacing the features at locations S with a background signal b. Then:*

$$\phi_{v_1}((h, w) \in N) = \frac{1}{HW} \sum_c GAP(x)_c (y_{chw} - b_{chw}) \tag{7.3}$$

Where GAP refers to global average pooling. We defer proof of this and other propositions to the Supplement. The results of this proposition mirrors the form of VEDML but with an added term to handle background distributions. These extra terms broaden the applicability of VEDML and we demonstrate their effect on explanations in Figure 7.3. In particular, we explain why two guitar players are similar in general (no background distribution), and relative to the second-best result of a guitar. Without a background, the explanation focuses on the guitar. However, when the explanation is relative to an image of a guitar the explanation focuses instead on the "tie-breaking" similarities, like the matching player. With counterfactual queries one can better understand a model's rationale behind *relative* similarity judgements and this can help in domains such as search engine optimization and automated medical diagnosis. We refer to Equation 7.3 as the Search Activation Map (SAM) in analogy with the Class Activation Map. We note that in non-GAP architectures, VEDML requires Taylor approximating nonlinear components. This heuristic corresponds estimating the Shapley values for a linear approximation of the true value function. For nonlinear architectures such as those that use cosine similarity, **SAM diverges from Shapley value theory** and hence violates its axioms. We can remedy this by using a Kernel-based Shapley value approximator [296] and refer to this approach as Kernel SAM.

Though the Shapley value framework unifies several methods for search engine interpretability, we note that the popular technique GradCAM does not align with Shapley value theory when applied to our feature-based value function (though it does align with Shapley values for GAP *classifiers*). To connect this approach to the theory of fair credit assignment, we show that GradCAM closely resembles Integrated Gradients (IG) [326], an approximator to the Aumann-Shapley values [327]:

Proposition 7.6.2. *Let $v(S) : [0,1]^N \to \mathbb{R} := f(mask(x, S))$ represent soft masking of the spatial locations of a deep feature map x with the vector of zeros and applying a differentiable function f. GradCAM is equivalent to Integrated Gradients approximated with a single sample at $\alpha = 1$ only if the function f has spatially invariant derivatives:*

$$\forall (h,w),(i,j) \in N : \frac{\partial f(x)}{\partial x_{chw}} = \frac{\partial f(x)}{\partial x_{cij}}$$

In typical case where f does not have spatially invariant derivatives ***GradCAM violates the dummy axiom*** *(see Section 7.4.2) and does not represent an approximation of Integrated Gradients.*

Where α refers to the parameter of IG that blends background and foreground samples. We note that the Aumann-Shapley values generalize the Shapley value to games where infinite numbers of players can join finitely many “coalitions”. These values align with Shapley values for linear functions but diverge in the nonlinear case. Proposition 7.6.2 also shows that in general GradCAM is sub-optimal and can be improved by considering Integrated Gradients on the feature space. We refer to this modification to GradCAM as Integrated Gradient Search Activation Maps or “IG SAM”. We also note that this modification can be applied to classifier-based GradCAM to yield a more principled classifier interpretation approach. We explore this and show an example of GradCAM violating the dummy axiom in the Supplement.

7.7 Second-Order Search Interpretations

Visualizing the pixels that explain a similarity judgement provides a simple way to inspect where a retrieval system is attending to. However, this visualization is only part of the story. Images can be similar for many different reasons, and a good explanation should clearly delineate these independent reasons. For example, consider the pair of images in the left column of Figure 7.6. These images show two similar scenes of people playing with dogs, but in different arrangements. We seek not just a heatmap highlighting similar aspects, but a data-structure capturing how parts of the query image *correspond* to parts of a retrieved image. To this end we seek to measure the interaction strength between areas of query and retrieved images as opposed to the effect of single features. We refer to this class of search and retrieval explanation methods as “second-order” methods due to their relation with second-order terms in the Shapley-Taylor expansion in Section 7.7.1.

7.7.1 Harsanyi Dividends

To capture the notion of interactions between query and retrieved images, we must consider credit assignments to *coalitions* of features. [293] formalize this notion with a unique and axiomatically specified way to assign credit or "Harsanyi Dividends" to every possible coalition, S, of N players in a cooperative game using the formula:

$$d_v(S) := \begin{cases} v(S) & \text{if } |S| = 1 \\ v(S) - \sum_{T \subsetneq S} d_v(T) & \text{if } |S| > 1 \end{cases} \tag{7.4}$$

These dividends provide a detailed view of the function's behavior at every coalition. In particular, [293] show that Shapley values arise from distributing these dividends evenly across members of the coalitions, a process we refer to a "projecting" the dividends down. In this work we seek a second-order analog of the Shapley values, so we generalize the notion of sharing these dividends between individuals to sharing these dividends between sub-coalitions. This computation re-derives the recently proposed Shapley-Taylor Indices [292], which generalize the Shapley values to coalitions of a size k using the discrete derivative operator. More specifically, by sharing dividends, we can alternatively express Shapley-Taylor values for coalitions $|S| = k$ as:

$$\phi_v^k(S) = \sum_{T:S \subset T} \frac{d_v(T)}{\binom{|T|}{|S|}} \tag{7.5}$$

Which states that the Shapley-Taylor indices arise from projecting Harsanyi dividends onto the k^{th} order terms. We note that this interpretation of the Shapley-Taylor indices is slightly more flexible than that of [292] as it allows one to define "jagged" fair credit assignments over just the coalitions of interest. Equipped with the Shapley-Taylor indices, ϕ_v^k, we can now formulate a value function for "second-order" search interpretations. As in the first order case, consider two spaces $\mathcal{X}$, $\mathcal{Y}$ equipped with a similarity function d. We introduce the second-order value function:

$$v_2(S) : 2^N \to \mathbb{R} := d(mask(x, S), mask(y, S)) \tag{7.6}$$

Where the grand coalition, $N = L_q \cup L_r$, are "locations" in both the query and retrieved images. These "locations" can represent either superpixels or coordinates in a deep feature map. Our challenge now reduces to computing Shapley-Taylor indices for this function.

7.7.2 A Fast Shapley-Taylor Approximation Kernel

Though the Harsanyi Dividends and Shapley-Taylor indices provide a robust way to allocate credit, they are difficult to compute. The authors of the Shapley-Taylor indices provide a sampling-based approximation, but this requires estimating each interaction term separately and scales poorly as dimensionality increases. To make this approach more tractable for high dimensional functions we draw a parallel to the unification of LIME with Shapley values through a linear regression weighting kernel. In particular, one can efficiently approximate Shapley values by randomly sampling coalitions, evaluating the value function, and fitting a

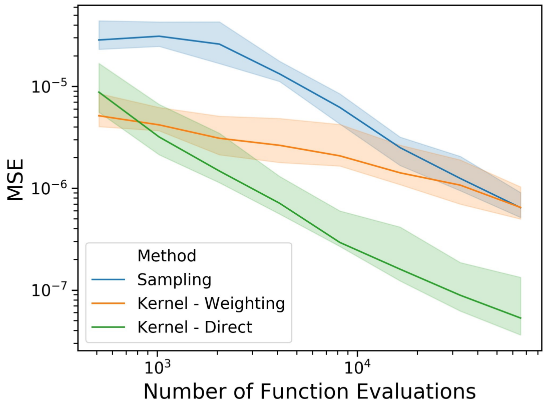


Figure 7.5: Convergence of Shapley-Taylor estimation schemes with respect to the Mean Squared Error (MSE) on randomly initialized deep networks with 15 dimensional input. Our strategies (Kernel) converge with significantly fewer function evaluations.

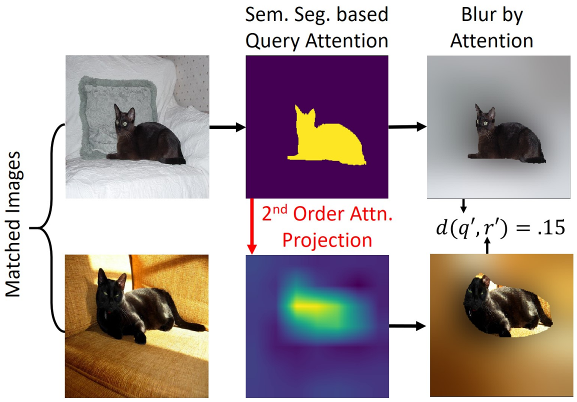


Figure 7.6: Our Second-order explanation evaluation strategy. A good method should project query objects (top left and middle) to corresponding objects in the retrieved image (bottom left and middle). When censoring all but these shared objects (right column) the search engine should view these images as similar.

weighted linear map from coalition vectors to function values. We find that this connection between Shapley values and weighted linear models naturally lifts to a weighted quadratic estimation problem in the "second-order" case. In particular, we introduce a weighting kernel for second order Shapley-Taylor indices:

$$\Lambda(S) = \frac{|N| - 1}{\binom{|N|}{|S|}\binom{|S|}{2}(|N| - |S|)} \tag{7.7}$$

Using this kernel, one can instead sample random coalitions, evaluate v, and aggregate the information into weighted quadratic model with a term for each distinct coalition $|S| \leq 2$. This allows one to approximate *all* Shapley-Taylor indices of $k = 2$ with a single sampling procedure, and often **requires** $10\times$ **fewer function evaluations to achieve the same estimation accuracy**. We show this speedup in Figure 7.5 on randomly initialized 15-dimensional deep networks. A detailed description of this and other experiments in this work are in the supplement. We find that one can further speed up the method by directly sampling from the induced distribution (Kernel-Direct) as opposed to randomly sampling coalitions and calculating weights (Kernel-Weighting). This direct sampling can be achieved by first sampling the size of the coalition from $p(s) \propto (|N| - 1)/(\binom{s}{2}(|N| - s))$ and then randomly sampling a coalition of that size. When our masking function operates on super-pixels, we refer to this as the second-order generalization of Kernel SHAP. This also gives insight into the proper form for a second-order generalization of LIME. In particular we add L1 regularization [328] and replace our kernel with a local similarity, $\Lambda(S) = \exp(-\lambda|mask(x, S); mask(y, S) - x; y|_2^2)$ where ";" represents concatenation, to create a higher-order analogue of LIME. Finally we note that certain terms of the kernel are undefined due to the presence of $\binom{s}{2}$ and $|N| - |S|$ in the denominator. These "infinite" weight terms encode hard constraints in the linear system and correspond to the efficiency axiom. In practice we enumerate these terms and give them a very large weight (10^8) in our regression. We reiterate that our kernel approximator converges to the same, uniquely-defined, values as prior sampling approaches but requires significantly fewer function evaluations.

7.7.3 Second-Order Search Activation Maps

In the first-order case, CAM and its search engine generalization, Search Activation Maps, arise naturally from the Shapley values of our first-order value function, Equation 7.2. To derive a second order generalization of SAM we now look to the Shapley-Taylor indices of our second order value function, Equation 7.6, applied to the same GAP architecture described in Proposition 7.6.1.

Proposition 7.7.1. *Let the spaces $\mathcal{X}$, $\mathcal{Y}$ and function d be as in Proposition 7.6.1. Let the grand coalition, N, index into the spatial coordinates of both the query image features x and retrieved image features y. Let the function $mask(y, S)$ act on a feature map y by replacing the corresponding features with a background feature map a for query features and b for retrieved features. Then:*

$$\phi_{v_2}(\{(h,w) \in \mathcal{L}_q, (i,j) \in \mathcal{L}_r\}) = \frac{1}{H^2W^2} \sum_c x_{chw}y_{cij} - a_{chw}y_{cij} - x_{chw}b_{cij} + a_{chw}b_{cij} \quad (7.8)$$

We note that the first term of the summation corresponds to the frequently used correlation layer [317–319, 329] and generalizes the "point-to-point" signal in [287]. In particular, **our axiomatically derived version has the extra terms allow counterfactual explanations against different background signals**. Like in the first-order case, this closed form only holds in the GAP architecture. To extend the method in a principled way we use our second-order kernel approximator and refer to this as second-order KSAM. We also introduce a generalization using a higher order analogue of Integrated Gradients, Integrated Hessians [324], applied to our feature maps. We refer to this as second-order IGSAM. In Section E.3 of the Supplement we prove that this approach is proportional to the Shapley-Taylor indices for the GAP architecture. We can visualize these second-order explanations by aggregating these Shapley-Taylor indices into a matrix with query image locations as rows and retrieved locations as columns. Using this matrix, we can "project" signals from a query to retrieved image. We show a few examples of attention projection using our second-order SAM in Figure 7.4.

7.8 Experimental Evaluation

First Order Evaluation Evaluating the quality of an interpretability method requires careful experimental design and is independent from what "looks good" to our human eye. If a model explanation method produces "semantic" connections between images it should be because to the underlying model is sensitive to these semantics. As a result, we adopt the evaluation strategy of [296], which measures how well the model explanation approximates the expected influence of individual features. In particular, these works calculate each feature's

Table 7.1: Comparison of performance of first- and second-order search explanation methods. Methods introduced in this work are highlighted in pink. *Though SAM generalizes [287] we refer to it as a baseline. For additional details see Section 7.8

Metric	Order	Model	SBSM	**PSHAP**	**LIME**	**KSHAP**	VESM	GCAM	SAM*	**IG SAM**	**KSAM**
			Model Agnostic				Architecture Dependent				
Faithfulness	First	DN121	0.18	**0.26**	0.23	0.24	0.08	0.12	0.12	**0.20**	**0.20**
		MoCoV2	0.22	**0.30**	0.28	**0.30**	0.13	0.19	0.21	0.25	**0.25**
		RN50	0.11	**0.16**	0.14	0.14	0.04	0.08	0.07	**0.11**	**0.11**
		VGG11	0.14	**0.16**	0.15	0.15	0.05	0.09	0.11	**0.14**	**0.14**
	Second	DN121	0.48	-	**0.54**	**0.54**	-	-	0.48	0.48	**0.49**
		MoCoV2	0.69	-	**0.74**	**0.74**	-	-	**0.72**	0.70	0.71
		RN50	0.74	-	**0.77**	**0.77**	-	-	**0.74**	**0.74**	**0.74**
		VGG11	0.68	-	**0.71**	**0.71**	-	-	0.69	0.69	**0.70**
Inefficiency	First	DN121	-	**0.00**	0.20	**0.00**	-	12.8	0.56	0.02	**0.00**
		MoCoV2	-	**0.00**	0.10	**0.00**	-	0.46	0.53	0.03	**0.00**
		RN50	-	**0.00**	0.22	**0.00**	-	14.9	0.47	0.03	**0.00**
		VGG11	-	**0.00**	0.27	**0.00**	-	4.20	0.54	0.05	**0.00**
	Second	DN121	-	-	0.14	**0.01**	-	-	0.21	0.03	**0.01**
		MoCoV2	-	-	0.13	**0.01**	-	-	0.20	0.02	**0.01**
		RN50	-	-	0.06	**0.01**	-	-	0.06	**0.01**	**0.01**
		VGG11	-	-	0.11	**0.01**	-	-	0.22	0.03	**0.01**
mIoU	Second	DN121	0.55	-	**0.68**	0.67	-	-	**0.68**	**0.68**	0.67
		MoCoV2	0.57	-	**0.70**	0.69	-	-	**0.70**	**0.70**	0.69
		RN50	0.55	-	**0.67**	0.66	-	-	**0.69**	0.66	0.65
		VGG11	0.54	-	**0.68**	0.67	-	-	0.72	**0.73**	0.70

importance, replace the top $n\%$ of features with background signal, and measure the effect on the function. A good model interpretability method should cause the replacement of the most important features, and hence cause the largest expected change in the function. We refer to this metric as the "Faithfulness" of an interpretation measure as it directly measures how well an interpretation method captures the behavior of an underlying model. Figure E.1 in the Supplement diagrams this process for clarity. In our experiments we blur the top 30% of image pixels to compute faithfulness. For those methods that permit it, we also measure how much the explanation violates the efficiency axiom. In particular we compare the sum of explanation coefficients with the value of $v(N) - v(\emptyset)$ and refer to this as the "Inefficiency" of the method. For additional details and evaluation code please see Section E.2 in the Supplement.

Second Order Evaluation In the second-order case we adopt the evaluation strategy of [324] which introduce a analogous second-order faithfulness measure. In particular, we measure how well model explanations approximate the expected *interaction* between two features. To achieve this, we select an object from the query image, use the second order explanation to find the corresponding object in the retrieved image, censor all but these two objects. We measure the new similarity as a measure of Faithfulness and illustrate this process in In Figure 7.6. We additionally quantify the inefficiency of several second-order methods as well as their effectiveness for semantic segmentation label propagation. In particular, we measure how well the explanation method can project a source object onto a target object. We treat this as a binary segmentation problem and measure the mean intersection over union (mIoU) of the projected object with respect to the true object mask. We note that mIoU is not a direct measurement of interpretation quality, but it can be useful for those intending to use model-interpretation methods for label propagation [144, 146]. These results demonstrate that axiomatically grounded model explanation methods such as IG SAM could offer improvement on downstream tasks. Because human evaluations introduce biases such as preference for compact or smoothness explanations, we consider Mechanical Turk [330] studies outside the scope of this work.

Datasets We evaluate our methods on the Pascal VOC [331] and MSCoCo [84] semantic segmentation datasets. To compute first and second order faithfulness we mine pairs of related images with shared object classes. We use the MoCo V2 [100] unsupervised image representation method to featurize the training and validation sets. For each image in the validation set we choose a random object from the image and find the training image that contains an object of the same class, a technique similiar to our evaluation strategy for CIR systems in Chapter 3 [332].

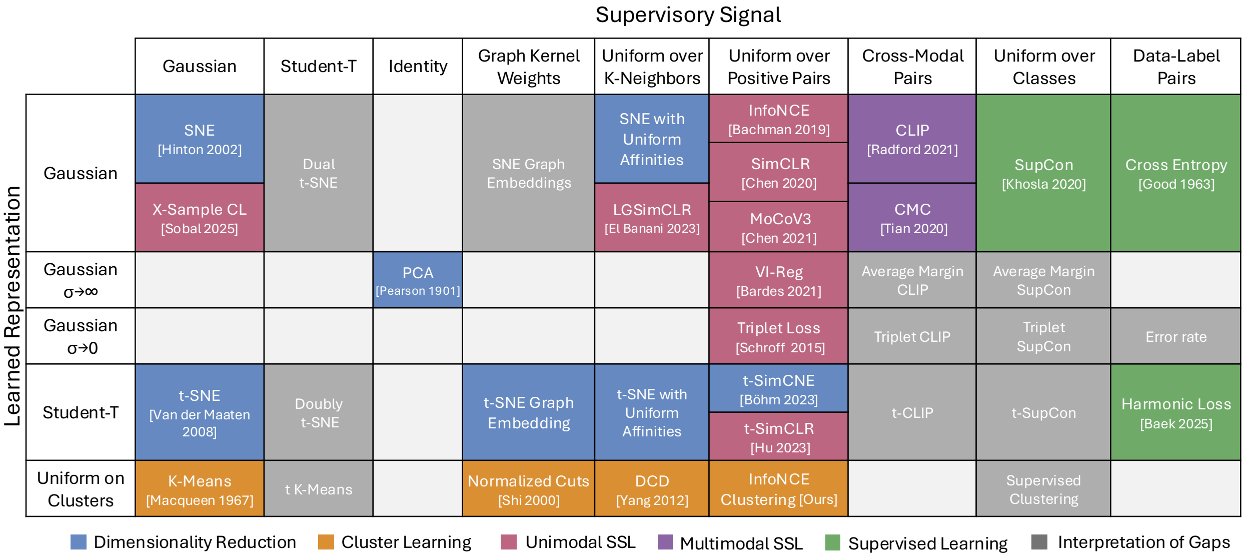
Results In Table 7.1 and Table E.3 of the Supplement we report experimental results for PascalVOC and MSCoCo respectively. We evaluate across visual search engines created from four different backbone networks: DenseNet121 [333], MoCo v2 [100], ResNet50 [121], and VGG11 [334] using cosine similarity on GAP features. As baselines we include VESM, SBSM, and SAM which generalizes [287]. We note that SBSM was not originally presented as a second-order method, and we describe how it can be lifted to this higher order setting in Section E.11 of the Supplement. We also evaluate several existing classifier explanation approaches applied to our search explanation value functions such as Integrated Gradients [335] on image pixels, Partition SHAP [325], LIME, Kernel SHAP (KSHAP), and GradCAM (GCAM) on deep feature maps [290]. For second-order variants of LIME and SHAP we used the local weighting kernel and our Shapley-Taylor approximation kernel from Section 7.7.2. Overall, several key trends appear. First, Shapley and Aumann-Shapley based approaches tend to be the most faithful and efficient methods, but at the price of longer computation time. One method that strikes a balance between speed and quality is our Integrated Gradient generalization of CAM which has both high faithfulness, low inefficiency, and only requires a handful of network evaluations ($\sim 10^2$). Furthermore, grey-box feature interpretation methods like SAM and IG SAM tend to perform better for label propagation. Finally, our methods beat existing baselines in several different categories and help to complete the space of higher order interpretation approaches. We point readers to the Section E.2 for additional details, compute information, and code.

7.9 Chapter Conclusion

In this work we have presented a uniquely specified and axiomatic framework for *model-agnostic* search, retrieval, and metric learning interpretability using the theory of Harsanyi dividends. We characterize search engine interpretability methods as either “first” or “second” order methods depending on whether they extract the most important areas or pairwise correspondences, respectively. We show that Shapley values of a particular class of value functions generalize many first-order methods, and this allows us to fix issues present in existing approaches and extend these approaches to counterfactual explanations. For second order methods we show that Shapley-Taylor indices generalize the work of [287] and use our framework to introduce generalizations of LIME, SHAP, and GradCAM. We apply these methods to extract image correspondences from opaque-box similarity models, a feat not yet presented in the literature. To accelerate estimation higher order Shapley-Taylor indices, we contribute a new weighting kernel that requires 10× fewer function evaluations. Finally, we show this game-theoretic formalism yields methods that are more “faithful” to the underlying model and better satisfy efficiency axioms across several visual similarity methods.

Chapter 8

I-Con: A Unifying Theory and Periodic Table of Representation Learning



Supervisory Signal

Learned Representation	Gaussian	Student-T	Identity	Graph Kernel Weights	Uniform over K-Neighbors	Uniform over Positive Pairs	Cross-Modal Pairs	Uniform over Classes	Data-Label Pairs
Gaussian	SNE [Hinton 2002] X-Sample CL [Sobal 2025]	Dual t-SNE		SNE Graph Embeddings	SNE with Uniform Affinities LGSimCLR [El Banani 2023]	InfoNCE [Bachman 2019] SimCLR [Chen 2020] MoCoV3 [Chen 2021]	CLIP [Radford 2021] CMC [Tian 2020]	SupCon [Khosla 2020]	Cross Entropy [Good 1963]
Gaussian σ→∞			PCA [Pearson 1901]			VI-Reg [Bardes 2021]	Average Margin CLIP	Average Margin SupCon	
Gaussian σ→0						Triplet Loss [Schroff 2015]	Triplet CLIP	Triplet SupCon	Error rate
Student-T	t-SNE [Van der Maaten 2008]	Doubly t-SNE		t-SNE Graph Embedding	t-SNE with Uniform Affinities	t-SimCNE [Böhm 2023] t-SimCLR [Hu 2023]	t-CLIP	t-SupCon	Harmonic Loss [Baek 2025]
Uniform on Clusters	K-Means [Macqueen 1967]	t K-Means		Normalized Cuts [Shi 2000]	DCD [Yang 2012]	InfoNCE Clustering [Ours]		Supervised Clustering	

🟦 Dimensionality Reduction 🟧 Cluster Learning 🟥 Unimodal SSL 🟪 Multimodal SSL 🟩 Supervised Learning ⬛ Interpretation of Gaps

Figure 8.1: **A "periodic" table of representation learning methods unified by the I-Con framework.** By choosing different types of conditional probability distributions over neighbors, I-Con generalizes over 23 commonly used representation learning methods.

8.1 Website and Video

For a quick video overview and blog post of this chapter, see https://mhamilton.net/icon.html

8.2 Chapter Summary

As the field of representation learning grows, there has been a proliferation of different loss functions to solve different classes of problems. We introduce a single information-theoretic equation that generalizes a large collection of modern loss functions in machine learning. In particular, we introduce a framework that shows that several broad classes of machine learning methods are precisely minimizing an integrated KL divergence between two conditional distributions: the supervisory and learned representations. This viewpoint exposes a hidden information geometry underlying clustering, spectral methods, dimensionality reduction, contrastive learning, and supervised learning. In previous chapters we have seen that the *relationships* between features are the key to discovering structure in complex systems. This chapter shows that this is not just a metaphor, but a deep unifying principle that cuts to the core of machine learning in general. This chapters theoretical framework enables the development of new loss functions by combining successful techniques from across the literature. We not only present a wide array of proofs, connecting over 23 different approaches, but we also leverage these theoretical results to create state-of-the-art unsupervised image classifiers that achieve a +8% improvement over the prior state-of-the-art on unsupervised classification on ImageNet-1K. We also demonstrate that I-Con can be used to derive principled debiasing methods which improve contrastive representation learners. Finally, using the machinery of this chapter we note that the algorithms of Chapters 4 and 6 fit neatly into our unified framework as "dense" generalizations of SimCLR and CLIP respectively.

8.3 Introduction

Over the past decade the field of representation learning has flourished, with new techniques, architectures, and loss functions emerging daily. These advances have powered state-of-the-art models in vision, language, and multimodal learning, often with minimal human supervision. Yet as the field expands, the diversity of loss functions makes it increasingly difficult to understand how different methods relate, and which objectives are best suited for a given task.

In this work, we introduce a general mathematical framework that unifies a wide range of representation learning techniques spanning supervised, unsupervised, and self-supervised approaches under a single information-theoretic objective. Our framework, **Information Contrastive Learning (I-Con)**, reveals that many seemingly disparate methods including clustering, spectral graph theory, contrastive learning, dimensionality reduction, and supervised classification are all special cases of the same underlying loss function.

While prior work has identified isolated connections between subsets of representation learning methods, typically linking only two or three techniques at a time [336–340], **I-Con is the first framework to unify over 23 distinct methods** under a single objective. This unified perspective not only clarifies the structure of existing techniques but also provides a strong foundation for transferring ideas and improvements across traditionally separate domains.

Using I-Con, we derive new unsupervised loss functions that significantly outperform previous methods on standard image classification benchmarks. Our key contributions are:

- We introduce *I-Con*, a single information-theoretic loss that generalizes several major classes of representation learning.

- We prove 15 theorems showing how diverse algorithms emerge as special cases of I-Con.

- We use I-Con to design a debiasing strategy that improves unsupervised ImageNet-1K accuracy by +**8%**, with additional gains of +**3%** on CIFAR-100 and +**2%** on STL-10 in linear probing.

8.4 Related Work

Representation learning spans a wide range of methods for extracting structure from complex data. We review approaches that I-Con builds upon and generalizes. For comprehensive surveys, see [21, 341, 342].

Feature Learning aims to derive informative low-dimensional embeddings using supervisory signals such as pairwise similarities, nearest neighbors, augmentations, class labels, or reconstruction losses. Classical methods like PCA [343] and MDS [344] preserve global structure, while UMAP [345] and t-SNE [346, 347] focus on local topology by minimizing divergences between joint distributions. I-Con adopts a similar divergence-minimization view.

Contrastive learning approaches such as SimCLR [348], CMC [261], CLIP [349], and MoCo v3 [350] use positive and negative pairs, often built via augmentations or aligned modalities. I-Con generalizes these losses within a unified KL-based framework, highlighting subtle distinctions between them. Supervised classifiers (e.g., ImageNet models [351]) also yield effective features, which I-Con recovers by treating class labels as discrete contrastive points, bridging supervised and unsupervised learning.

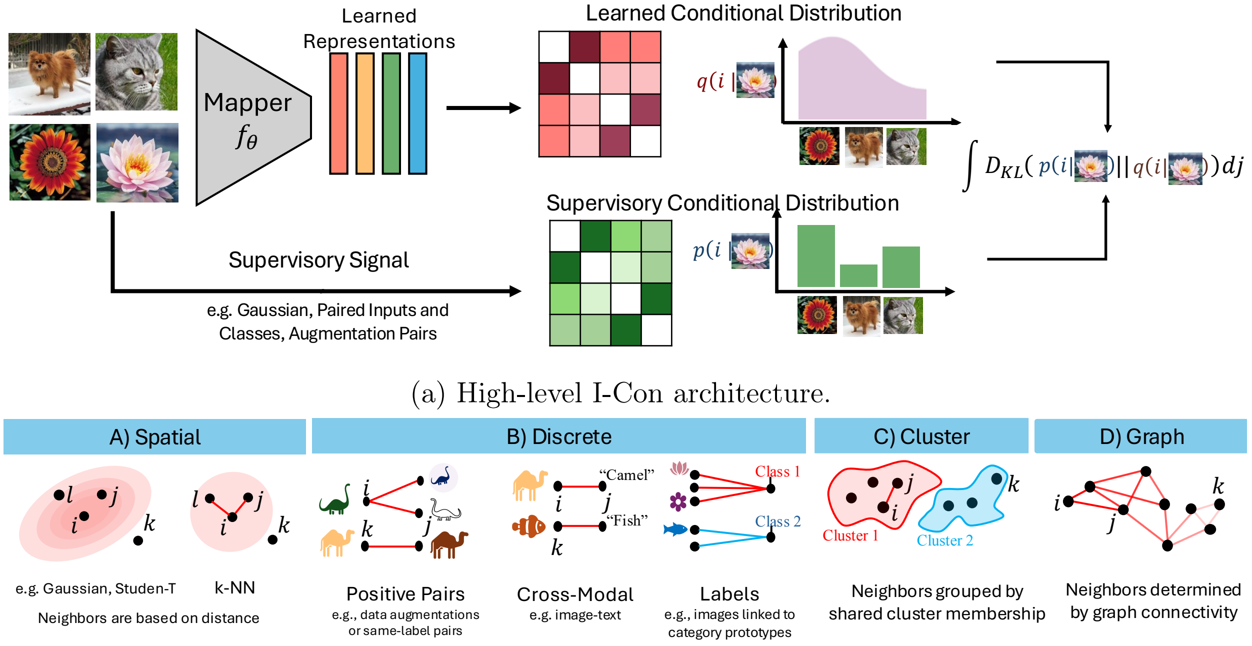
Clustering methods uncover discrete structure through distance metrics, graph partitions, or contrastive supervision. Algorithms like k-Means [352], EM [353], and spectral clustering [354] are foundational. Recent methods, including IIC [250], Contrastive Clustering [355], and SCAN [356], leverage invariance and neighborhood structure. Teacher-student models such as TEMI [357] and EMA-based architectures [133] enhance clustering further. I-Con encompasses these by aligning a clustering-induced joint distribution with a target distribution derived from similarity, structure, or contrastive signals.

Unifying Representation Learning has been explored through connections between contrastive learning and t-SNE [337, 339], equivalences between contrastive and cross-entropy losses [338], and relations between spectral and contrastive methods [336, 340]. Other efforts, like Bayesian grammar models [358], offer probabilistic perspectives. Tschannen et al. [359] emphasized estimator and architecture design in mutual information frameworks but stopped short of broader unification.

While prior work links subsets of these methods, I-Con, to our knowledge, is the first to unify supervised, contrastive, clustering, and dimensionality reduction objectives under a single loss. This perspective clarifies their shared structure and opens paths to new learning principles.

8.5 Methods

The I-Con framework unifies multiple representation learning methods under a single loss function: minimizing the average KL divergence between two conditional "neighborhood distributions" that define transition probabilities between data points. This information-


Learned Representations
Mapper f_θ
Learned Conditional Distribution
$q(i|$
Supervisory Conditional Distribution
$p(i|$
$\int D_{KL}(p(i|\;)||q(i|\;))dj$
Supervisory Signal
e.g. Gaussian, Paired Inputs and Classes, Augmentation Pairs
A) Spatial
B) Discrete
C) Cluster
D) Graph
l
j
i
k
e.g. Gaussian, Studen-T
k-NN
Neighbors are based on distance
Positive Pairs
e.g., data augmentations or same-label pairs
"Camel"
"Fish"
Cross-Modal
e.g. image-text
Class 1
Class 2
Labels
e.g., images linked to category prototypes
Cluster 1
Cluster 2
Neighbors grouped by shared cluster membership
Neighbors determined by graph connectivity

(a) High-level I-Con architecture.

(b) Illustrative examples of distribution families for p_θ or q_ϕ.

Figure 8.2: **Overview of the I-Con framework**. (a) Alignment of learned and supervisory distributions. (b) Common distribution families in I-Con's formulation.

theoretic objective generalizes techniques from clustering, contrastive learning, dimensionality reduction, spectral graph theory, and supervised learning. By varying the construction of the supervisory distribution and the learned distribution, I-Con encompasses a broad class of existing and novel methods. We introduce I-Con and demonstrate its ability to unify techniques from diverse areas and orchestrate the transfer of ideas across different domains, leading to a state-of-the-art unsupervised image classification method.

8.5.1 Information Contrastive Learning

Let $i, j \in \mathcal{X}$ be elements of a dataset $\mathcal{X}$, with a probabilistic neighborhood function $p(j|i)$ defining a transition probability. To ensure valid probability distributions, $p(j|i) \geq 0$ and $\int_{j\in\mathcal{X}} p(j|i) = 1$. We parameterize this distribution by $\theta \in \Theta$, to create a learnable function $p_\theta(j|i)$. Similarly, we define another distribution $q_\phi(j|i)$ parameterized by $\phi \in \Phi$. The core I-Con loss function is then:

$$\mathcal{L}(\theta, \phi) = \int_{i\in\mathcal{X}} D_{\mathrm{KL}}\left(p_\theta(\cdot|i)||q_\phi(\cdot|i)\right) = \int_{i\in\mathcal{X}}\int_{j\in\mathcal{X}} p_\theta(j|i)\log\frac{p_\theta(j|i)}{q_\phi(j|i)}. \tag{8.1}$$

In practice, p is typically a fixed "supervisory" distribution, while q_ϕ is learned by comparing deep network representations, prototypes, or clusters. Figure 8.2a illustrates this alignment process. The optimization aligns q_ϕ with p, minimizing their KL divergence. Although most existing methods optimize only q_ϕ, I-Con also allows learning both p_θ and q_ϕ, although one must take care to prevent trivial solutions.

8.5.2 Unifying Representation Learning Algorithms with I-Con

Despite the incredible simplicity of Equation 8.1, this equation is rich enough to generalize several existing methods in the literature simply by choosing parameterized neighborhood distributions p_θ and q_ϕ as shown in Figure 8.1. We categorize common choices for p_θ and q_ϕ in Figure 8.2a.

Table 8.1 summarizes some key choices which recreate popular methods from contrastive learning (SimCLR, MOCOv3, SupCon, CMC, CLIP, VICReg), dimensionality reduction (SNE, t-SNE, PCA), clustering (K-Means, Spectral, DCD, PMI), and supervised learning (Cross-Entropy and Harmonic Loss). Due to limited space, we defer proofs of each of these theorems to the supplemental material. We also note that Table 8.1 is not exhaustive, and we encourage the community to explore whether other learning frameworks implicitly minimize Equation 8.1 for some choice of p and q.

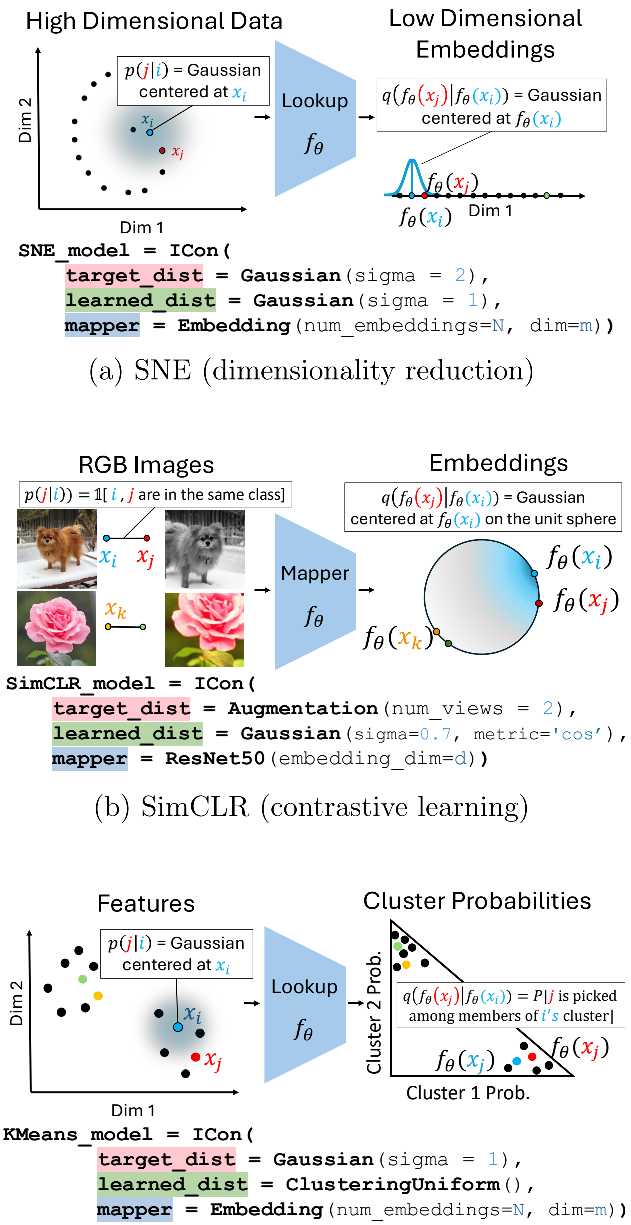
Example: SNE, SimCLR, and K-Means

While I-Con unifies a broad range of methods, we illustrate how different choices of p and q recover well-known techniques such as SNE, SimCLR, and K-Means. Full details are in the appendix.

SNE as "neighbors remain neighbors." Stochastic Neighbor Embedding (SNE) is a classic example. Given $x \in \mathbb{R}^{d \times n}$ with n points in d dimensions, SNE learns a low-dimensional representation $\phi \in \mathbb{R}^{m \times n}$, typically $m \ll d$. To preserve local structure, $p(j \mid i)$ is defined by placing a Gaussian around each high-dimensional point x_i, and $q_\phi(j \mid i)$ by placing a Gaussian around ϕ_i. Minimizing the average KL divergence between these distributions ensures that points close in the original space remain close in the embedded space.

SimCLR as "augmentations of the same image are neighbors." Contrastive learning methods like SimCLR and SupCon instead use class labels. Here, $p(j \mid i) = 1$ if j is an augmentation of i (and 0 otherwise). In the embedding space, $q_\phi(j \mid i)$ is defined via a Gaussian-like distribution based on cosine similarity. Minimizing their KL divergence encourages images from the same scene to cluster together.

K-Means as "points that are close are members of the same clusters." Clustering-based approaches like K-Means and DCD follow a similar recipe. The distribution $p(j \mid i)$ is again Gaussian-based in the original space, while $q_\phi(j \mid i)$ reflects whether points are assigned to the same cluster in the learned representation. Minimizing KL divergence aligns these cluster assignments with the actual neighborhood structure in the data. Methods like K-Means include an entropy penalty to enforce hard probabilistic assignments, as shown in Theorem F.4.2, whereas methods like DCD do not include it.


High Dimensional Data
Dim 2
Dim 1
$p(j|i)$ = Gaussian centered at x_i
x_i
x_j
Lookup
f_θ
Low Dimensional Embeddings
$q(f_\theta(x_j)|f_\theta(x_i))$ = Gaussian centered at $f_\theta(x_i)$
$f_\theta(x_j)$
$f_\theta(x_i)$
Dim 1

```
SNE_model = ICon(
    target_dist = Gaussian(sigma = 2),
    learned_dist = Gaussian(sigma = 1),
    mapper = Embedding(num_embeddings=N, dim=m))
```

(a) SNE (dimensionality reduction)

RGB Images
$p(j|i)) = \mathbb{1}[\,i, j$ are in the same class]
x_i
x_j
x_k
Mapper
f_θ
Embeddings
$q(f_\theta(x_j)|f_\theta(x_i))$ = Gaussian centered at $f_\theta(x_i)$ on the unit sphere
$f_\theta(x_i)$
$f_\theta(x_j)$
$f_\theta(x_k)$

```
SimCLR_model = ICon(
    target_dist = Augmentation(num_views = 2),
    learned_dist = Gaussian(sigma=0.7, metric='cos'),
    mapper = ResNet50(embedding_dim=d))
```

(b) SimCLR (contrastive learning)

Features
Dim 2
Dim 1
$p(j|i)$ = Gaussian centered at x_i
x_i
x_j
Lookup
f_θ
Cluster Probabilities
Cluster 2 Prob.
Cluster 1 Prob.
$q(f_\theta(x_j)|f_\theta(x_i)) = P[j$ is picked among members of $i's$ cluster]
$f_\theta(x_j)$
$f_\theta(x_j)$

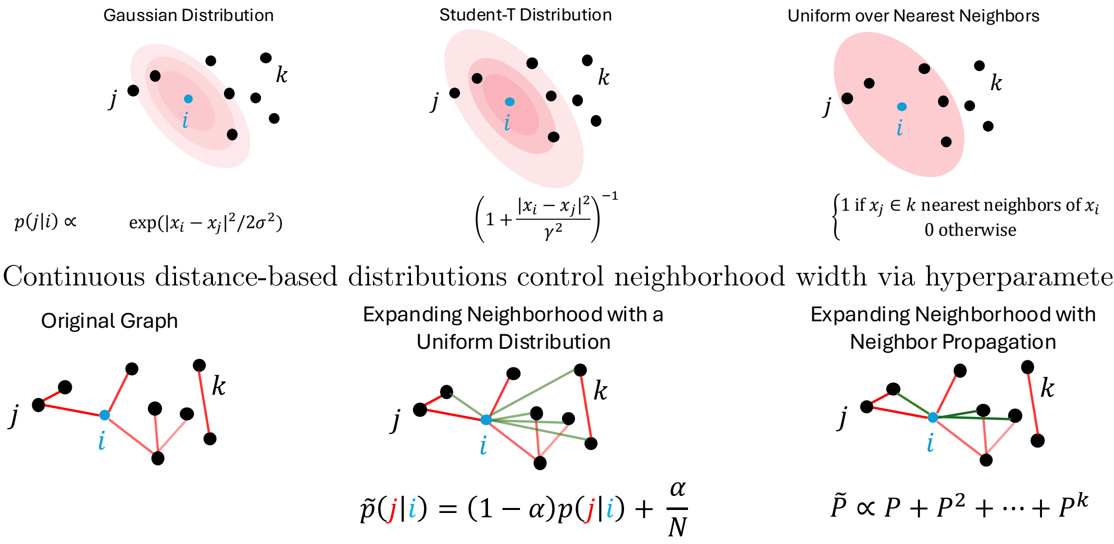
```
KMeans_model = ICon(
    target_dist = Gaussian(sigma = 1),
    learned_dist = ClusteringUniform(),
    mapper = Embedding(num_embeddings=N, dim=m))
```

(c) K-Means (clustering)

Figure 8.3: Examples of methods as special cases of I-Con via different choices of p and q, with corresponding code-style configurations.

Method	**Choice of $p_\theta(j \mid i)$**	**Choice of $q_\phi(j \mid i)$**
(A) Dimensionality Reduction		
SNE [346] Theorem F.2.1	Gaussian over data points, x_i $\frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$	Gaussian over learned low-dimensional points, ϕ_i $\frac{\exp(-\|\phi_i - \phi_j\|^2)}{\sum_{k \neq i} \exp(-\|\phi_i - \phi_k\|^2)}$
t-SNE [347] Corollary 1		Cauchy distribution over ϕ_i $\frac{(1 + \|\phi_i - \phi_j\|^2)^{-1}}{\sum_{k \neq i}(1 + \|\phi_i - \phi_k\|^2)^{-1}}$
PCA [343] Theorem F.2.2	$\mathbb{1}[\, i = j \,]$	Wide Gaussian on linear projection features, $f_\phi(x_i)$ $\lim_{\sigma \to \infty} \frac{\exp(-\|f_\phi(x_i) - f_\phi(x_j)\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|f_\phi(x_i) - f_\phi(x_k)\|^2/2\sigma^2)}$
(B) Contrastive Learning		
InfoNCE Loss [360] Theorem F.3.1	$\frac{1}{Z}\mathbb{1}[i$ and j are a positive pair$]$	Gaussian on deep normalized features $\frac{\exp\big(f_\phi(x_i) \cdot f_\phi(x_j)\big)}{\sum\limits_{k \neq i} \exp\big(f_\phi(x_i) \cdot f_\phi(x_k)\big)}$
Triplet Loss [361] Theorem F.3.3		Gaussian on deep features (1 neg. sample, $\sigma \to 0$) $\frac{\exp\big(-\|f_\phi(x_i) - f_\phi(x_j)\|^2/2\sigma^2\big)}{\sum\limits_{k \in \{i^+, i^-\}} \exp\big(-\|f_\phi(x_i) - f_\phi(x_k)\|^2/2\sigma^2\big)}$
t-SimCLR, t-SimCNE [337, 339] Corollary 2		Student-T on deep features $\frac{(1 + \|\phi_i - \phi_j\|^2/\nu)^{-(\nu+1)/2}}{\sum_{k \neq i}(1 + \|\phi_i - \phi_k\|^2/\nu)^{-(\nu+1)/2}}$
VICReg* without covariance term [362] Theorem F.3.2		Wide Gaussian on learned features $\lim_{\sigma \to \infty} \frac{\exp(-\|f_\phi(x_i) - f_\phi(x_j)\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|f_\phi(x_i) - f_\phi(x_k)\|^2/2\sigma^2)}$
SupCon [363] Theorem F.3.4	$\frac{1}{Z}\, \mathbb{1}[i$ and j have same class$]$	Gaussian on deep normalized features $\frac{\exp\big(f_\phi(x_i) \cdot f_\phi(x_j)\big)}{\sum\limits_{k \neq i} \exp\big(f_\phi(x_i) \cdot f_\phi(x_k)\big)}$
X-Sample [336] Theorem F.3.5	Gaussian on corresponding embeddings $\frac{\exp\big(g_\theta(x_i) \cdot g_\theta(x_j)\big)}{\sum\limits_{k \neq i} \exp\big(g_\theta(x_i) \cdot_\theta (x_k)\big)}$	
LGSimCLR [364]	$\frac{1}{Z}\, \mathbb{1}[x_i$ is among x_j's k nearest neighbors$]$	
CMC & CLIP [261] Theorem F.3.6	$\frac{1}{Z}\, \mathbb{1}[i,j$ pos. pairs, $V_i \neq V_j]$	$\frac{\exp\big(f_\phi(x_i) \cdot f_\phi(x_j)\big)}{\sum_{k \in V_j} \exp\big(f_\phi(x_i) \cdot f_\phi(x_k)\big)}$
(C) Supervised Learning		
Supervised Cross Entropy [365] Theorem F.3.7	Indicator over classes $\mathbb{1}\big[i$ belongs to class $j\big]$	$\frac{\exp\big(f_\phi(x_i) \cdot \phi_j\big)}{\sum_{k \in C} \exp\big(f_\phi(x_i) \cdot \phi_k\big)}$
Harmonic Loss [366] Theorem F.3.8		Student-T on deep features and class prototypes $\lim_{\sigma \to 0} \frac{(\sigma^2 + \|f_\phi(x_i) - \phi_j\|^2)^{-n}}{\sum_{k \in C}(\sigma^2 + \|f_\phi(x_i) - \phi_k\|)^{-n}}$
Masked Lang. Modeling [367] Theorem F.3.9	$\frac{1}{Z}\, \#\big[$Context i precedes token $j\big]$	$\frac{\exp\big(f_\phi(x_i) \cdot \phi_j\big)}{\sum_{k \in C} \exp\big(f_\phi(x_i) \cdot \phi_k\big)}$
(D) Clustering		
Probabilistic k-Means [352] Theorem F.4.2	Intra-cluster uniform probability $\sum\limits_{c=1}^{m} \frac{p\big(f_\theta(x_i) \text{ and } f_\theta(x_j) \text{ in } c\big)}{\mathbb{E}[\text{size of cluster } c]}$	Gaussians on datapoints $\frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$
Spectral Clustering [368] Corollary 4		Gaussians on spectral embeddings $\frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$
Normalized Cuts [354] Theorem F.4.3	Intra-cluster uniform probability weighted by degree $\sum\limits_{c=1}^{m} \frac{p\big(f_\theta(x_i) \text{ and } f_\theta(x_j) \text{ in } c\big) \cdot d_j}{\mathbb{E}[\text{degree of cluster } c]}$	Gaussians on graph weigths $\frac{\exp(w_{ij}/d_j)}{\sum_k \exp(w_{ik}/d_k)}$
PMI Clustering [357] Theorem F.4.4	$\frac{1}{k}\, \mathbb{1}[j$ is k-NN of $i]$	Intra-Cluster Uniform Probability $\sum\limits_{c=1}^{m} \frac{p\big(f_\theta(x_i) \text{ and } f_\theta(x_j) \text{ in } c\big)}{\mathbb{E}[\text{size of cluster } c]}$
Debaised InfoNCE Clustering (ours)	Debiased Graph through Uniform Distribution and Neighbor Propagation	$\sum\limits_{c=1}^{m} \frac{(1-\alpha)p\big(f_\theta(x_i) \text{ and } f_\theta(x_j) \text{ in } c\big)}{\mathbb{E}[\text{size of cluster } c]} + \frac{\alpha}{N}$

Table 8.1: **I-Con unifies representation learners** under different choices of $p_\theta(j|i)$ and $q_\phi(j|i)$. Proofs of the propositions in this table can be found in the supplement.


Gaussian Distribution
Student-T Distribution
Uniform over Nearest Neighbors
j
i
k
$p(j|i) \propto$
$\exp(|x_i - x_j|^2/2\sigma^2)$
$\left(1 + \frac{|x_i - x_j|^2}{\gamma^2}\right)^{-1}$
$\begin{cases} 1 \text{ if } x_j \in k \text{ nearest neighbors of } x_i \\ 0 \text{ otherwise} \end{cases}$
Original Graph
Expanding Neighborhood with a Uniform Distribution
Expanding Neighborhood with Neighbor Propagation
$\tilde{p}(j|i) = (1-\alpha)p(j|i) + \frac{\alpha}{N}$
$\tilde{P} \propto P + P^2 + \cdots + P^k$

(a) Continuous distance-based distributions control neighborhood width via hyperparameters.

(b) Graph-based distributions expand neighborhoods through structural strategies.

Figure 8.4: **Neighborhood adaptation in continuous and discrete settings.** (a) Distance-based distributions modulate neighborhood "width" via parameters such as σ. (b) Graph-based approaches modify the connectivity directly, often via random walks or added edges, thereby broadening each node's neighborhood.

8.5.3 Creating New Representation Learners with I-Con

The I-Con framework unifies various approaches to representation learning under a single mathematical formulation and, crucially, facilitates the transfer of techniques among different domains. For instance, a trick from contrastive learning can be applied to clustering—or vice versa. In this chapter, we demonstrate how surveying modern representation methods enables the development of clustering and unsupervised classification algorithms that surpass previous performance levels. Specifically, we integrate insights from spectral clustering, t-SNE, and debiased contrastive learning [369] to build a state-of-the-art unsupervised image classification pipeline.

Debasing

Debiased Contrastive Learning (DCL) addresses the mismatch caused by random negative sampling in contrastive learning, especially when the number of classes is small. Randomly chosen negatives can turn out to be positives, introducing spurious repulsive forces between similar examples. [369] rectify this by subtracting out such false repulsion terms and boosting attractive forces, substantially improving representation quality. However, their method modifies the softmax itself, implying that $q_{j|i}$ is no longer a genuine probability distribution and making it more difficult to extend the approach to clustering or supervised tasks.

Our view, grounded in the I-Con framework, suggests a simpler and more general alternative: rather than adjusting the learned distribution $q_{j|i}$, we incorporate additional "uncertainty" directly into the supervisory distribution $p(j|i)$. This preserves $q_{j|i}$ as a valid distribution and keeps the method applicable to tasks beyond contrastive learning.

Debiasing through a Uniform Distribution

Our first example adopts a simple uniform mixture:

$$\tilde{p}(j|i) \ = \ (1-\alpha)\, p(j|i) \ + \ \frac{\alpha}{N},$$

where N is the local neighborhood size, and α specifies the degree of mixing. This approach assigns a small probability mass $\frac{\alpha}{N}$ to each "negative" sample, thereby mitigating overconfident allocations. In supervised contexts, this is analogous to label smoothing [370]. In contrast, [369] adjust the softmax function itself while retaining one-hot labels.

Another way to view this method is through the lens of heavier-tailed or broader distributions. By adding a uniform component, we mirror the idea in t-SNE's Student-t distribution [347], which allocates greater mass to distant points. In both cases, expanding the distribution reduces the chance of overfitting to a narrowly defined set of neighbors.

Empirical results in Tables 8.3, Figures 8.5, and 8.6 show that this lightweight modification consistently improves performance across various tasks and batch sizes. It also "relaxes" overconfident distributions, much like label smoothing in supervised cross entropy, thereby guarding against vanishing gradients.

Debiasing through Neighbor Propagation

A second strategy applies graph-based expansions. As shown in Table 8.1, replacing k-Means' Gaussian neighborhoods with degree-weighted k-nearest neighbors recovers spectral clustering, which is known for robust, high-quality solutions. Building on this idea, we train contrastive learners with KNN-based neighborhood definitions. Given the nearest-neighbor graph, we can further expand it by taking longer walks, analogous to Word-Graph2Vec or tsNET [371, 372], a process we term *neighbor propagation*.

Formally, let P be the conditional distribution matrix whose entries $P_{ij} = p(x_j \mid x_i)$ define the probability of selecting x_j as a neighbor of x_i. Interpreting P as the adjacency matrix of the training data, we can smooth it by summing powers of P up to length k:

$$\tilde{P} \ \propto \ P + P^2 + \cdots + P^k.$$

We can further simplify this by taking a uniform distribution over all points reachable within k steps, denoted by:

$$\tilde{P}_U \ \propto \ I\big[\, P + P^2 + \cdots + P^k \ > \ 0 \big],$$

where $I[\cdot]$ is the indicator function. This walk-based smoothing broadens the effective neighborhood, allowing the model to learn from a denser supervisory signal.

Tables 8.3 and 8.4 confirm that adopting such a propagation-based approach yields significant improvements in unsupervised image classification, underscoring the effectiveness of neighborhood expansion as a debiasing strategy.

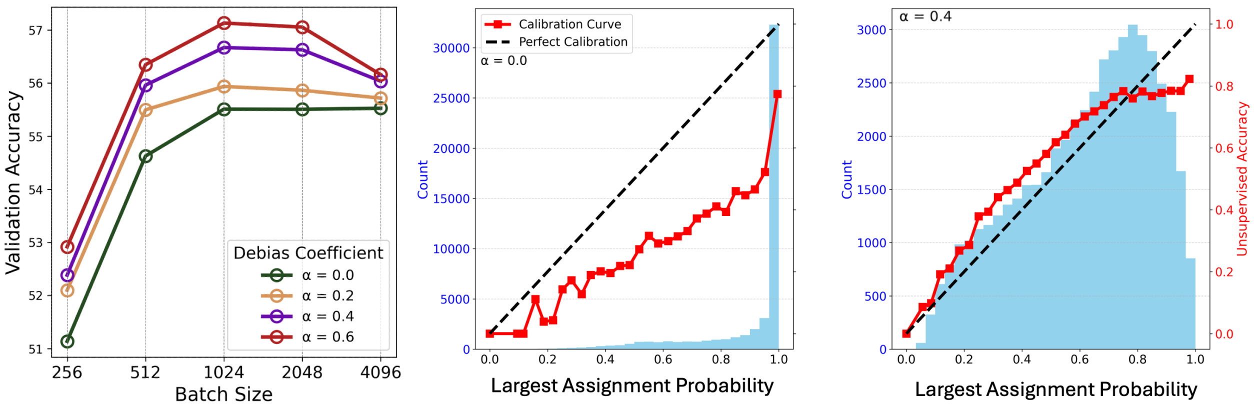


Figure 8.5: Left: Debiasing cluster learning improves performance on ImageNet-1K across batch sizes. Center: Distribution of maximum predicted probabilities for the biased model ($\alpha = 0$) showing poor calibration, with overconfident predictions. Right: Distribution of maximum predicted probabilities for the debiased model ($\alpha = 0.4$), demonstrating improved probability calibration. Debiased training alleviates optimization stiffness by reducing the prevalence of saturated logits, mitigating vanishing gradient issues, and fostering more robust and well-calibrated learning dynamics.

8.6 Experiments

In this section, we demonstrate that the I-Con framework offers testable hypotheses and practical insights into self-supervised and unsupervised learning. Rather than aiming only for state-of-the-art performance, our goal is to show how I-Con can enhance existing unsupervised learning methods by leveraging a unified information-theoretic approach. Through this framework, we also highlight the potential for cross-pollination between techniques in varied machine learning domains, such as clustering, contrastive learning, and dimensionality reduction. This transfer of techniques, enabled by I-Con, can significantly improve existing methodologies and open new avenues for exploration.

We focus our experiments on clustering because it is relatively understudied compared to contrastive learning, and there are a variety of techniques that can now be adapted to this task. By connecting established methods such as k-Means, SimCLR, and t-SNE within the I-Con framework, we uncover a wide range of possibilities for improving clustering methods. We validate these theoretical insights experimentally, demonstrating the practical impact of I-Con.

We evaluate the I-Con framework using the ImageNet-1K dataset [35], which consists of 1,000 classes and over one million high-resolution images. This dataset is considered one of the most challenging benchmarks for unsupervised image classification due to its scale and complexity. To ensure a fair comparison with prior works, we strictly adhere to the experimental protocol introduced by [357]. The primary metric for evaluating clustering performance is Hungarian accuracy, which measures the quality of cluster assignments by finding the optimal alignment between predicted clusters and ground truth labels via the Hungarian algorithm [250]. This approach provides a robust measure of clustering performance in an unsupervised context, where direct label supervision is absent during training.

For feature extraction, we utilize the DiNO pre-trained Vision Transformer (ViT) models in three variants: ViT-S/14, ViT-B/14, and ViT-L/14 [373]. These models are chosen to ensure comparability with previous work and to explore how the I-Con framework performs across varying model capacities. The experimental setup, including training protocols, optimization strategies, and data augmentations, mirrors those used in TEMI to ensure consistency in methodology.

The training process involved optimizing a linear classifier on top of the features extracted by the DiNO models. Each model was trained for 30 epochs, using ADAM [374] with a batch size of 4096 and an initial learning rate of 1e-3. We decayed the learning rate by a factor of 0.5 every 10 epochs to allow for stable convergence. We do not apply additional normalization to the feature vectors. During training, we applied a variety of data augmentation techniques, including random re-scaling, cropping, color jittering, and Gaussian blurring, to create robust feature representations. Furthermore, to enhance the clustering performance, we pre-computed global nearest neighbors for each image in the dataset using cosine similarity. This allowed us to sample two augmentations and two nearest neighbors for each image in every training batch, thus incorporating both local and global information into the learned representations. We refer to our derived approach as “InfoNCE Clusting” in Table 8.2. In particular, we use a supervisory neighborhood comprised of augmentations, KNNs ($k = 3$), and KNN walks of length 1. We use the “shared cluster likelihood by cluster” neighborhood from k-Means (See table 8.1 for a more detailed Equation) as our learned neighborhood function to drive cluster learning.

8.6.1 Baselines

We compare our method against several state-of-the-art clustering methods, including TEMI, SCAN, IIC, and Contrastive Clustering. These methods rely on augmentations and learned representations, but often require additional regularization terms or loss adjustments, such as controlling cluster size or reducing the weight of affinity losses. In contrast, our I-Con-based loss function is self-balancing and does not require such manual tuning, making it a cleaner, more theoretically grounded approach. This allows us to achieve higher accuracy and more stable convergence across three different-sized backbones.

8.6.2 Results

Table 8.2 compared the Hungarian accuracy of Debiased InfoNCE Clustering across different DiNO variants (ViT-S/14, ViT-B/14, ViT-L/14) and several other modern clustering methods. The I-Con framework consistently outperforms the prior state-of-the-art method across all model sizes. Specifically, for the DiNO ViT-B/14 and ViT-L/14 models, debiased InfoNCE clustering achieves significant performance gains of +4.5% and +7.8% in Hungarian accuracy compared to TEMI, the prior state-of-the-art ImageNet clusterer. We attribute these improvements to two main factors:

Self-Balancing Loss: Unlike TEMI or SCAN, which require hand-tuned regularizations (e.g., balancing cluster sizes or managing the weight of affinity losses), I-Con’s loss function automatically balances these factors without additional regularization hyper-parameter

Method	DiNO ViT-S/14	DiNO ViT-B/14	DiNO ViT-L/14
k-Means	51.84	52.26	53.36
Contrastive Clustering	47.35	55.64	59.84
SCAN	49.20	55.60	60.15
TEMI	56.84	58.62	–
InfoNCE Clust. (Ours)	**57.8** ± 0.26	**64.75** ± 0.18	**67.52** ± 0.28

Table 8.2: Comparison of methods on ImageNet-1K clustering with respect to Hungarian Accuracy. Debiased InfoNCE Clustering significantly outperforms the prior state-of-the-art TEMI. Note that TEMI does not report results for ViT-L.

Method	DiNO ViT-S/14	DiNO ViT-B/14	DiNO ViT-L/14
Baseline	55.51	63.03	65.70
+ Debiasing	57.27± 0.07	63.72 ± 0.09	66.87 ± 0.07
+ KNN Propagation	**58.45** ± 0.23	64.87 ± 0.19	67.25 ± 0.21
+ EMA	57.8 ± 0.26	**64.75** ± 0.18	**67.52** ± 0.28

Table 8.3: Ablation study of new techniques discovered through the I-Con framework. We compare ImageNet-1K clustering accuracy across different sized backbones.

tuning as we are using the exact same clustering kernel used by k-Means. This theoretical underpinning leads to more robust and accurate clusters.

Cross-Domain Insights: I-Con leverages insights from contrastive learning to refine clustering by looking at pairs of images based on their embeddings, treating augmentations and neighbors similarly. This approach, originally successful in contrastive learning, translates well into clustering and leads to improved performance on noisy high-dimensional image data.

8.6.3 Ablations

We conduct several ablation studies to experimentally justify the architectural improvements that emerged from analyzing contrastive clustering through the I-Con framework. These ablations focus on two key areas: the effect of incorporating debiasing into the target and embedding spaces and the impact of neighbor propagation strategies.

We perform experiments with different levels of debiasing in the target distribution, denoted by the parameter α, and test configurations where debiasing is applied to the target side, both sides (target and learned representations), or none. As seen in Figure 8.6, adding debiasing improves performance, with the optimal value typically around $\alpha = 0.6$ to $\alpha = 0.8$, particularly when applied to both sides of the learning process. This method is similar to how debiasing work in contrastive learning by assuming that each negative sample has a non-zero probability (α/N) of being incorrect. Figure 8.5 shows how changing the value of α improves performance across different batch sizes.

In a second set of experiments, shown in Table 8.4, we examine the impact of neighbor propagation strategies. We evaluate clustering performance when local and global neighbors

Method	DiNO ViT-S/14	DiNO ViT-B/14	DiNO ViT-L/14
Baseline	55.51	63.03	65.72
+ KNNs	56.43	64.26	65.70
+ 1-walks on KNN	**58.09**	**64.29**	65.97
+ 2-walks on KNN	57.84	64.27	**67.26**
+ 3-walks on KNN	57.82	64.15	67.02

Table 8.4: Ablation Study on Neighbor Propagation. Adding both KNNs and walks of length 1 or 2 on the KNN graph achieves the best performance.

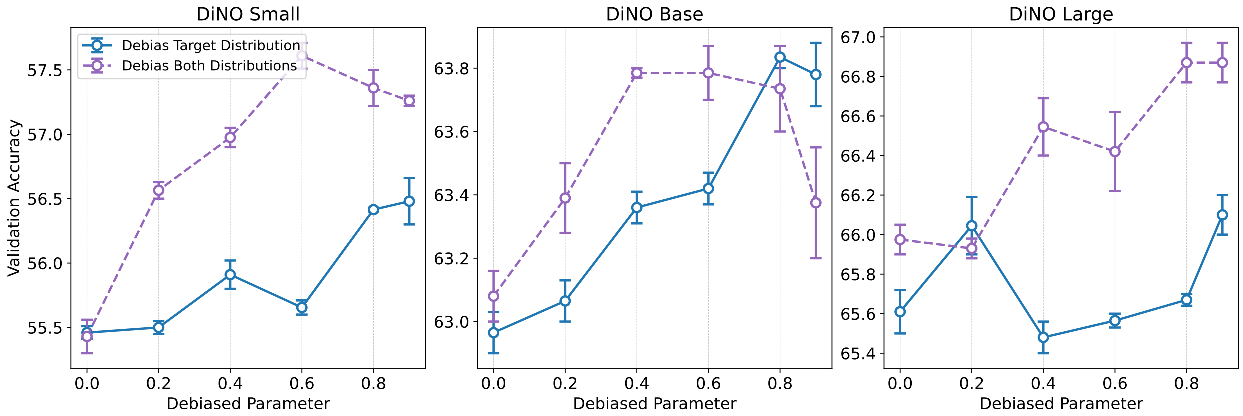


Figure 8.6: Effects of increasing the debias weight α on the supervisory neighborhood (blue line) and both the learned and supervisory neighborhood (red line). Adding some amount of debiasing helps in all cases, with a double debiasing yielding the largest improvements.

are included in the contrastive loss computation. Neighbor propagation, especially at small scales ($s = 1$ and $s = 2$), significantly boosts performance across all model sizes, showing the importance of capturing local structure in the embedding space. Larger neighbor propagation values (e.g., $s = 3$) offer diminishing returns, suggesting that over-propagating neighbors may dilute the information from the nearest, most relevant points. Note that only DiNO-L/14 showed preference for large step size, and this is likely due to its higher k-nearest neighbor ability, so the augmented links are correct.

Our ablation studies highlight that small adjustments in the debiasing parameter and neighbor propagation can lead to notable improvements that achieve a state-of-the-art result with a simple loss function. Additionally, sensitivity to α and propagation size varies across models, with larger models generally benefiting more from increased propagation but requiring fine-tuning of α for optimal performance. We recommend using $\alpha \approx 0.6$ to $\alpha \approx 0.8$ and limiting neighbor propagation to small values for a balance between performance and computational efficiency.

8.7 Chapter Conclusion

In summary, we have developed I-Con: a single information-theoretic equation that unifies a broad class of machine learning methods. We provide over 15 theorems that prove this assertion for many of the most popular loss functions used in clustering, spectral graph theory, supervised and unsupervised contrastive learning, dimensionality reduction, and supervised classification and regression. We not only theoretically unify these algorithms but show that our connections can help us discover new state-of-the-art methods, and apply improvements discovered for a particular method to any other method in the class. We illustrate this by creating a new method for unsupervised image classification that achieves a +8% improvement over prior art. We believe that the results presented in this work represent just a fraction of the methods that are potentially unify-able with I-Con, and we hope the community can use this viewpoint to improve collaboration and analysis across algorithms and machine learning disciplines.

Chapter 9

Conclusion

9.1 Retrospective on the Central Hypothesis

This thesis aimed to understand how to create algorithms that can discover structure in complex systems without human guidance. Through building systems that can learn without our guidance, we inch closer to automating the discovery of new scientific knowledge, allowing us to see further into the unknown than ever before. Each work in this thesis highlights a simple—but far-reaching—idea: **many rich structures of the world can be recovered *without human guidance* by analyzing *relationships* between self-supervised representations**. In particular, Chapters 3-6 showed that studying the relationships between deep representations can yield novel and state-of-the-art approaches to:

- Trace artistic motifs across millennia and media (Chapter 3) by finding close pairs of representations that span these gaps.
- Discover "blind-spots" in generative algorithms where they fail to model the data (Chapter 3) by analyzing nodes in representation retrieval data-structures.
- Discover visual objects and classify every pixel of the world without human supervision (Chapter 4) by distilling relationships between representations into a semantic segmentation system.
- Improve the resolution of any model's representations by 64× while retaining their precise semantics (Chapter 5) by analyzing how representations change when we apply small transformations to the input image.
- Rediscover the meaning of words in a language and the location of sounds (Chapter 6 by comparing dense representations in a contrastive loss.

Not only do we show that relationships between deep representations have significant practical use, but they provide a powerful theoretical hammer to unify broad swatch of different techniques both across model explainability and representation learning. These theoretical analyses of Chapters 7 and 8 show how these relationships can discover fine-grained semantics about the natural world, and how this notion of considering relationships between representations can be used to unify over 23 algorithms across the field of machine earning. More specifically, Chapter 7, shows that inner products between deep network representations approximate the “Shapley-Taylor Index”, the unique axiomatically-determined way to distribute credit for a model’s predictions over its inner representations. This observation not only yields new algorithms to better explain predictions from search engines and other retrieval systems, but gives us an understanding for why these comparisons between representations have the power to “pull blood from the stone” and directly localize information even when they were never trained to do so. Finally, Chapter 8 shows that relationships between features form the basis for a single equation that unifies over 20 commonly used learning objectives into a periodic table of machine learning (Chapter 8). This periodic table not only unifies many algorithms across the field, but predicts the existence of fundamentally new algorithms that can learn without human labels. We show that one-such predicted algorithm yields a new state-of-the-art in unsupervised image classification.

9.2 Future Work

The unifying thread of this thesis has been the pursuit of algorithms and frameworks that allow us to extract interpretable structure, generate new hypotheses, and automate key steps in scientific discovery—often in domains where human intuition and ground truth are limited or inaccessible. Looking ahead, future work will aim to expand the reach and depth of these approaches: by extending algorithmic discovery to new and challenging real-world systems, by leveraging advances in large language models to automate broader aspects of the scientific process, and by further developing unifying theoretical frameworks like I-Con that map out the landscape of machine learning itself. Together, these directions promise to not only deepen our understanding of complex phenomena, but also to accelerate the pace of discovery across scientific disciplines.

9.2.1 Decoding Vocalization of the Atlantic Spotted Dolphin

A core motivation throughout this thesis is the pursuit of algorithms that can reveal interpretable, meaningful structure in complex and poorly understood systems, even in the absence of explicit human supervision or ground truth. This vision naturally extends to one of the most challenging and intriguing frontiers for unsupervised discovery: decoding non-human communication systems.

In ongoing collaborations with marine biologists from the Wild Dolphin Project, we are applying and adapting the self-supervised and multimodal learning methods developed in this thesis to the domain of Atlantic spotted dolphin communication. The available data—roughly 50 hours of synchronized underwater audio and video recordings—presents a uniquely rich but challenging environment, marked by the scarcity of labeled data, the subtlety of behavioral cues, and the fundamentally different structure of dolphin communication compared to human language.

Initial experiments transferring dense audio-visual matching techniques like those of Chapter 6 to this domain revealed significant challenges: models trained solely on dolphin data performed at chance, highlighting the limits of data efficiency and the differences in

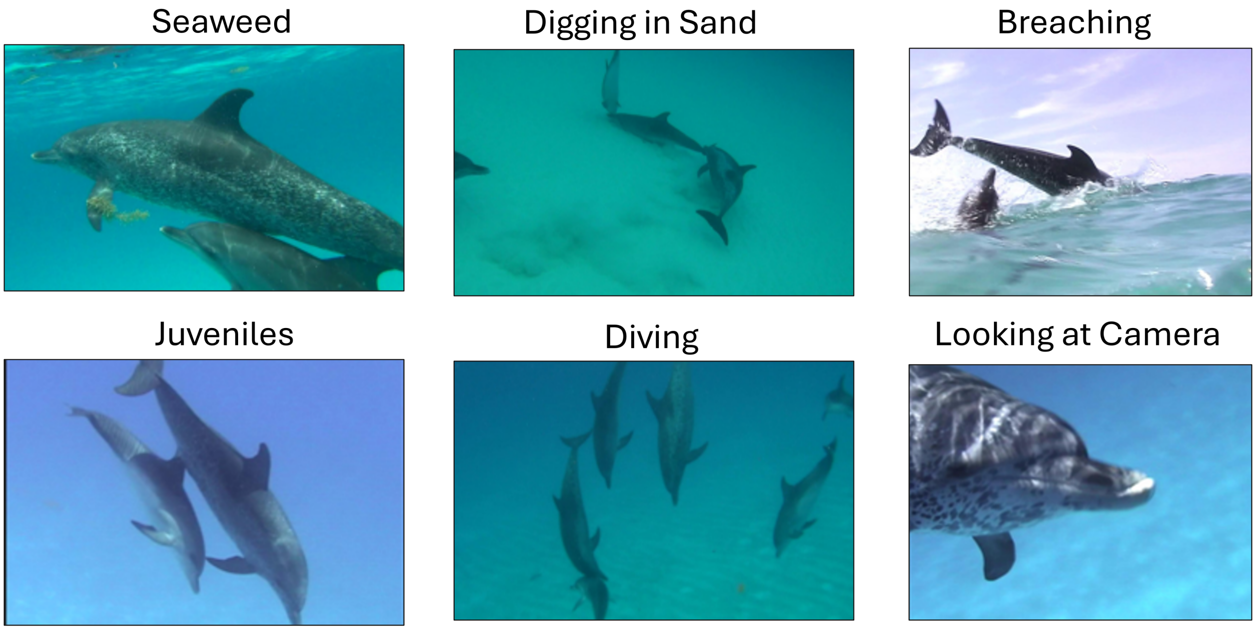


Figure 9.1: Examples of LLM Spotted Dolphin Behavior from the hours of the Wild Dolphin Project Dataset

communicative context. To address the lack of extensive human annotation, we are now leveraging large language models to generate high-coverage semantic labels for each video frame, spanning dozens of behavioral and environmental attributes. Examples of these LLM annotations can be seen in Figure 9.1

By reframing the dolphin audio-visual matching task as a contrastive learning problem against this rich set of automatically generated semantic labels, early results show that the model can distinguish between key types of dolphin vocalizations (such as whistles and clicks) and achieve performance well above chance. Visualizations of the learned representations, like those shown in Figure 9.2, suggest that the models are beginning to organize dolphin signals according to functionally meaningful behavioral categories—a first step toward identifying potential "words" or motifs in dolphin communication.

This ongoing work epitomizes the thesis's central theme: using modern AI to probe the structure of complex natural phenomena without prior human understanding. The path forward involves deeper interpretability analysis, integration of temporal and multimodal information, and close collaboration with domain experts to validate and refine emergent communication hypotheses. Ultimately, these efforts may not only illuminate the "language" of dolphins, but also serve as a testbed for developing truly general-purpose algorithms for scientific discovery in domains where ground truth is fundamentally inaccessible.

9.2.2 Toward Automating Science with Large Language Models

The central theme of this thesis is the development of algorithms that automate aspects of scientific discovery, ranging from unsupervised extraction of interpretable structure in complex datasets to the generation of new hypotheses in domains where ground truth is

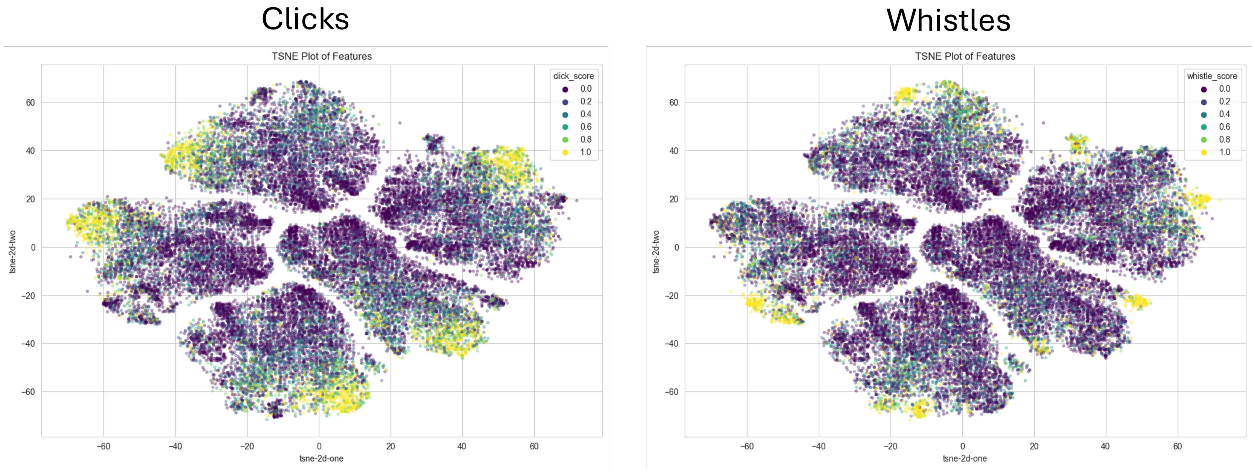


Figure 9.2: Visualization of deep audio representations of dolphin communication colored by communication type. Deep features have clearly learned to separate echolocation clicks from whistles in an emergent manner.

unknown. This perspective naturally leads to a broader, emerging question: to what extent can artificial intelligence systems automate not only pattern-finding, but the very process of scientific inquiry itself?

Recent advances in large language models (LLMs), such as GPT-4, have brought this vision closer to reality. Core components of scientific reasoning—including literature synthesis, data extraction, and hypothesis formation—are increasingly within reach of automation. In ongoing collaborations, we are investigating the use of LLMs to transform labor-intensive scientific workflows. In particular, we are conducting studies to evaluate whether LLMs can largely automate key steps of systematic literature review in environmental health, a field where the timely and rigorous synthesis of evidence is vital for public decision-making.

These collaborations focus on building semi-automated review pipelines in which LLMs are deployed to screen abstracts and full texts, extract structured data, and provide transparent justifications for their selections. Early results suggest that these systems can match or exceed human performance in tasks such as abstract screening—achieving AUC/AP values above 96% and offering dramatic savings in expert labor. Furthermore, LLMs are demonstrating promising accuracy in information extraction tasks, laying the groundwork for scientific reviews that are faster, more reproducible, and more auditable than traditional manual approaches.

This direction of research forms a natural continuum with the rest of this thesis: from building algorithms that automate the discovery of structure in raw data, to automating the broader process of scientific reasoning itself. Looking ahead, we aim to extend these approaches beyond literature review to encompass ideation, experiment design, and hypothesis generation, ultimately exploring how AI systems can augment and accelerate the entire scientific process.

9.2.3 Extending I-Con, Our Unifying Framework for Representation Learning

A central contribution of this thesis is the unification of a wide array of machine learning algorithms—ranging from clustering and dimensionality reduction to contrastive and spectral methods—under the I-Con (Information-Contrastive) framework. By expressing these diverse approaches as special cases of a single underlying objective, we introduce a "periodic table" of machine learning that provides both a conceptual map of the field and a practical guide for discovering new algorithms. A natural avenue for future work is the systematic exploration and filling of the gaps in this table. Many rows and columns correspond to as-yet-unstudied combinations of supervision type and representation structure. By exploring these gaps and empirically validating the resulting methods, we may uncover new approaches with desirable properties for unsupervised learning, transfer, and interpretability—much as the periodic table of elements historically predicted the existence of new chemical species before their discovery.

Another promising direction lies in re-examining the mathematical structure of the I-Con objective itself. The present framework relies heavily on the Kullback-Leibler (KL) divergence as the measure of discrepancy between distributions. However, there is no fundamental reason to restrict attention to this single divergence or metric. Exploring alternative distances—such as Wasserstein, Jensen-Shannon, or energy-based divergences—could yield unifying views that extend beyond current applications, potentially connecting the I-Con framework to new domains such as optimal transport, generative modeling, or robust representation learning. Such generalizations may not only provide deeper theoretical insight but also lead to the discovery of algorithms better suited to specific data modalities or learning tasks, revealing new organizing principles in machine learning.

Finally, the current periodic table is built upon relatively simple mathematical objects: vectors, clusters, and (to some extent) graph structures or tokens. Yet, scientific data and learning problems often involve richer and more structured entities, including sets, functions, manifolds, trees, or even entire programs. Expanding the I-Con framework to incorporate these higher-order objects would require defining new forms of representation and appropriate notions of similarity or "friendship." This could enable the periodic table to accommodate domains such as program synthesis, symbolic reasoning, or heirarchical methods. This would further advancing the thesis's central ambition: to create algorithmic tools capable of uncovering structure and regularity in even the most complex and underexplored systems. By extending the I-Con approach along these lines, we open the door to both a deeper theoretical synthesis of machine learning and powerful new engines for scientific discovery.

9.3 Closing Remarks

Johannes Kepler, working by candlelight, distilled elliptical orbits from painstaking tables of star positions; Dmitri Mendeleev glimpsed the periodicity of the elements by sorting index cards on a train ride. Their breakthroughs came not from labeled data sets, but from relentless scrutiny of *relationships* in the data. This thesis argues and empirically demonstrates that modern machine learning can and should emulate this spirit.

By elevating relationships to first-class citizens in our algorithms, we have shown that machines can *see* without bounding boxes, *hear* without knowledge of text, and more generally *discover* taxonomies and classifications of information without human guidance. We show that focusing on the relationships is not just a metaphor, but rather an observation with the capability to unify the fields of both model explainability and representation learning.

The road ahead is long, but the guiding principle is clear: If we wish our models to uncover new science, new art, and new understanding, we must continue to look not at the isolated points but at the invisible threads that bind them together.

Appendix A

Appendix for Chapter 3

A.1 Visualizing Failure Cases

Figure A.1 (a) shows how conditioners that do not share a common support can yield low diversity conditional neighbors. Though sharing a common support is certainly helpful, it is not mandatory as shown by Figure A.1 (b). Some potential mitigations for these effects could be to fine tune learned embeddings to promote diverse queries, or to re-weight query outputs based on diversity. Additionally, an initial alignment with an optimal transport method could mitigate these effects [375].

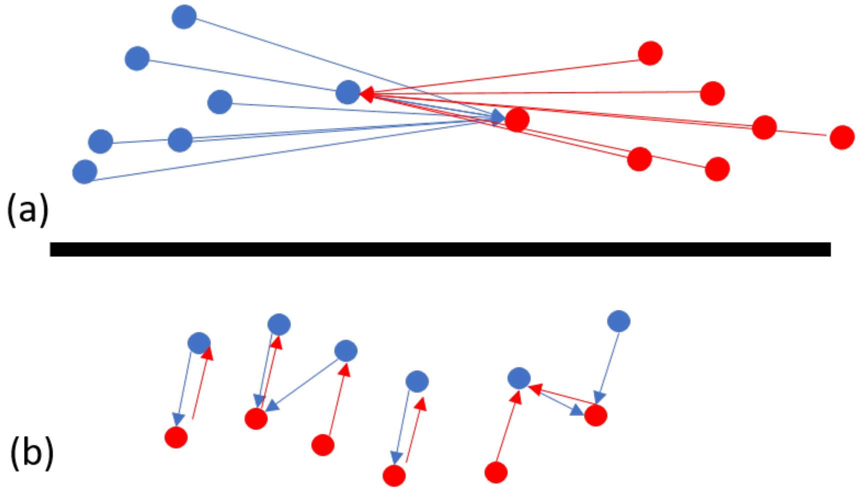


Figure A.1: A schematic illustration of how conditional KNN can yield to a lack of diversity in particular geometries. (a) shows how low diversity can occur when there is no overlap of supports. Figure (b) shows how support intersection is not necessary for quality alignment

A.2 Additional Matches

In addition to the matches displayed in Figure 3.1 we provide several additional results. Figure A.2 shows additional matches for a single query, and Figure A.3 shows matches across several different queries. Figure A.4 shows random matches to give a sense of the method's average-case results.

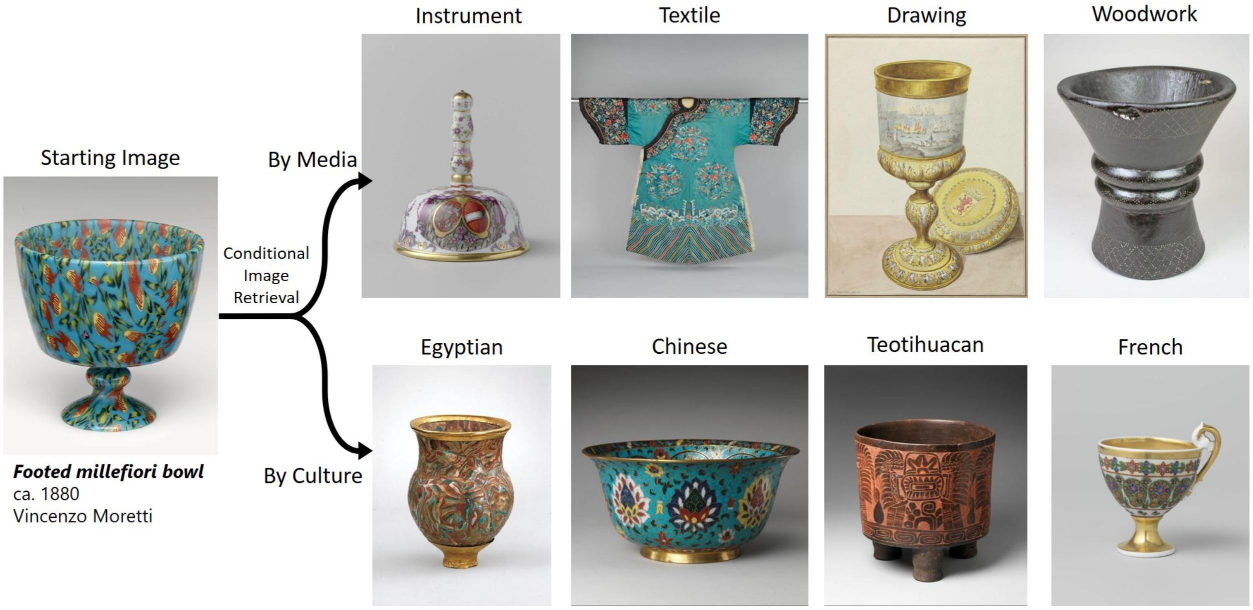


Figure A.2: Additional conditional image retrieval results on artworks from the Metropolitan Museum of Art and Rijksmuseum using media (top row text) and culture (bottom row text) as conditioners.

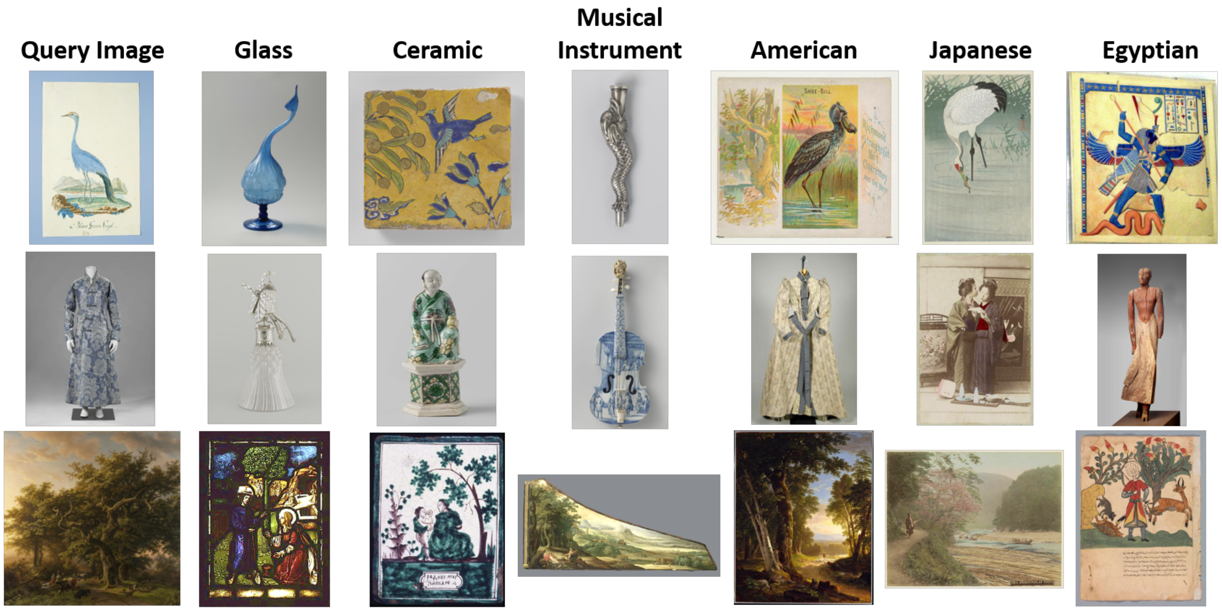


Figure A.3: Additional conditional image retrieval results on artworks from the Metropolitan Museum of Art and Rijksmuseum using top row text as conditioners.

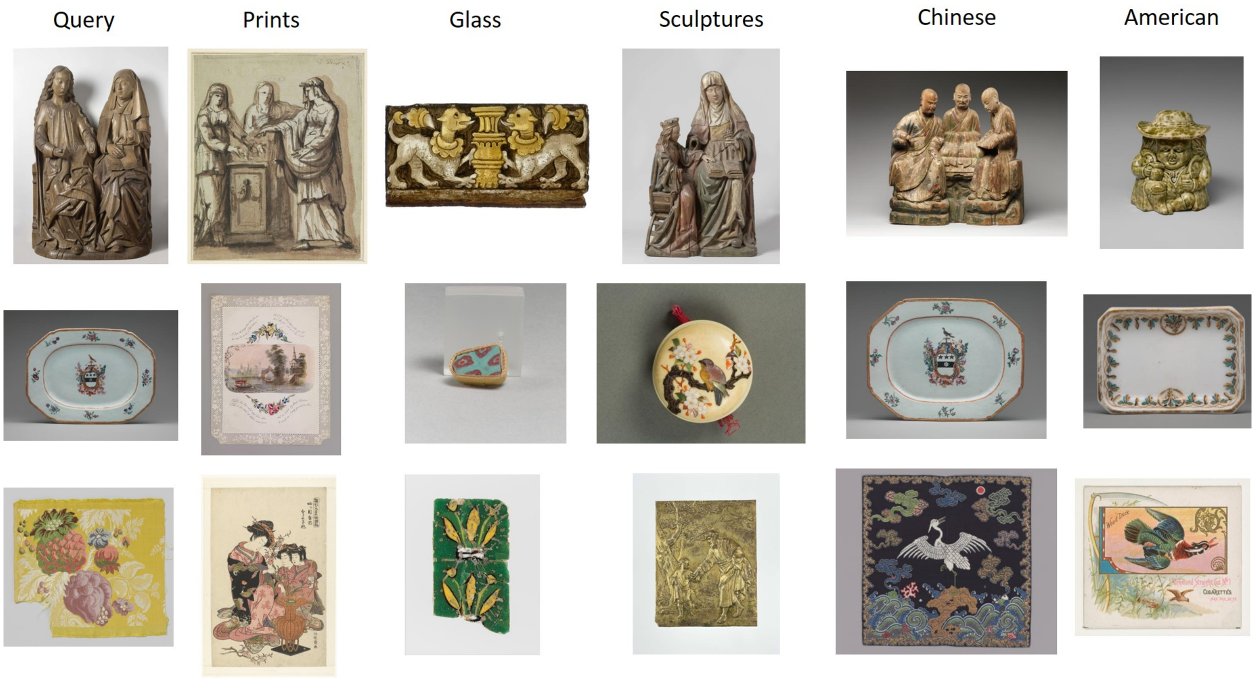


Figure A.4: Randomly selected conditional image retrieval results on artworks from the Metropolitan Museum of Art and Rijksmuseum using top row text as conditioners.

A.3 Proof of Theorem 1

In the following analysis, suppose an RPTree-Max is built using a dataset $\mathcal{X} \subset \mathbb{R}^D$, of diameter W, with doubling dimension $\leq d$. Furthermore, assume that the size reduction rate at any given level of the tree is bounded above by γ.

Lemma A.3.1. *For any ball, B of radius $R > 0$ and any $0 < \epsilon < 1$, there exists a constant $c_1 > 0$ such that with probability $> 1 - \epsilon$, B will be completely inscribed inside of an RPTree-Max cell of radius no more than $c_1 R d \sqrt{d} \log(d)$*

Proof. We modify the proof of Theorem 12 from [51]. In particular we let $\Delta^* = \frac{1}{\epsilon} c_5 R d \sqrt{d} \log(d)$, where c_5 refers to the constant of Lemma 11 of [51] The rest of the proof proceeds without modification. □

Lemma A.3.2. *For any finite set of balls, $\{B_i\}$, with constant radii $R > 0$, and any $0 < \epsilon < 1$, there exists a constant $c_2 > 0$ such that with probability $> 1 - \epsilon$, every B_i will be completely inscribed inside of an RPTree-Max cell of radius no more than $c_2 R d \sqrt{d} \log(d)$*

Proof. We proceed by induction on the number of balls. Lemma A.3.1 provides the base case of $|\{B_i\}| = 1$. For the inductive case we assume the lemma holds for a set $\{B_i\}$ of size n, with $\epsilon' = \frac{\epsilon}{8}$ and constant c_2'. Given an additional B_{n+1}, we can leverage our base case to select an $\epsilon'' = \frac{\epsilon}{8}$ and constant c_2''. We can see that the probability that both events occur simultaneously is bounded above by:

$$(1 - \epsilon')(1 - \epsilon'') = (1 - \frac{\epsilon}{8})(1 - \frac{\epsilon}{8}) = 1 - \frac{\epsilon}{4} - \frac{\epsilon^2}{64} < 1 - \epsilon$$

Finally, using the new constant, $c_2 = \max(c_2', c_2'')$, the radii criterion holds for all balls. □

Theorem A.3.3. *(Restatement of Theorem 1) Suppose an RPTree-Max, $\mathcal{T}$, is built using a dataset $\mathcal{X} \subset \mathbb{R}^D$, of diameter W, with doubling dimension $\leq d$. Further suppose $\mathcal{T}$ is balanced with a cell-size reduction rate bounded above by γ. Let $\mathcal{S} \subseteq \mathcal{X}$ be a subset of the dataset used to build the tree and $\mathcal{B}$ a finite set of radius $R > 0$ balls that cover $\mathcal{S}$. For every $0 < \epsilon < 1$ there exists a constant, $c > 0$, such that with probability $> 1 - \epsilon$ the fraction of cells that contain points within $\mathcal{S}$ is bounded above by $|\mathcal{B}| 2^{-log_\gamma(W/R')}$ where $R' = cRd\sqrt{d}\log(d)$*

Proof. We begin by invoking Lemma A.3.2, which shows that each ball of our covering will end up completely inscribed within small radii cells of $\mathcal{T}$. For each ball we upper bound their contribution to the total fraction of cells that contain points within $\mathcal{S}$.

Consider any ball $B_i \in \mathcal{B}$ in the covering. By Lemma A.3.2 we know this ball is inscribed within a cell of radius R'. Our goal is to show that this cell must be several levels down in the tree. By our regularity conditions we know that at each subsequent level of a tree, the cell size decreases by at most a factor of γ. So to achieve the reduction in size from W to R', the cell must lie at or below level $\log_\gamma(W/R')$. At worst, every child of our cell contains a point within $\mathcal{S}$. Because $\mathcal{T}$ is balanced, the ratio of cell children to total cells of the tree is at most $2^{-log_\gamma(W/R')}$. At worst each ball of the cover, $\mathcal{B}$, is in a separate branch of the tree so combining these contributions yields $|\mathcal{B}| 2^{-log_\gamma(W/R')}$.

□

Appendix B

Appendix for Chapter 4

B.1 Video and Code

We include a short video description of our work at https://aka.ms/stego-video.

We also provide training and evaluation code at https://aka.ms/stego-code

B.2 Additional Results on the Potsdam-3 Dataset

In addition to our evaluations in Section 4.6.1 we compare STEGO to prior art on the Potsdam 3-class aerial image segmentation task presented in [93]. In Table B.1 We find that STEGO is able to achieve +12% accuracy compared to the previous state of the art, IIC. We show example qualitative results in Figure B.1.

Table B.1: Additional results on the Potsdam-3 aerial image segmentation challenge

Model	**Unsup. Acc.**
Random CNN [93]	38.2
K-Means [58]	45.7
SIFT [123]	38.2
[124]	49.6
[125]	63.9
Deep Cluster [122]	41.7
IIC [93]	65.1
STEGO (Ours)	**77.0**

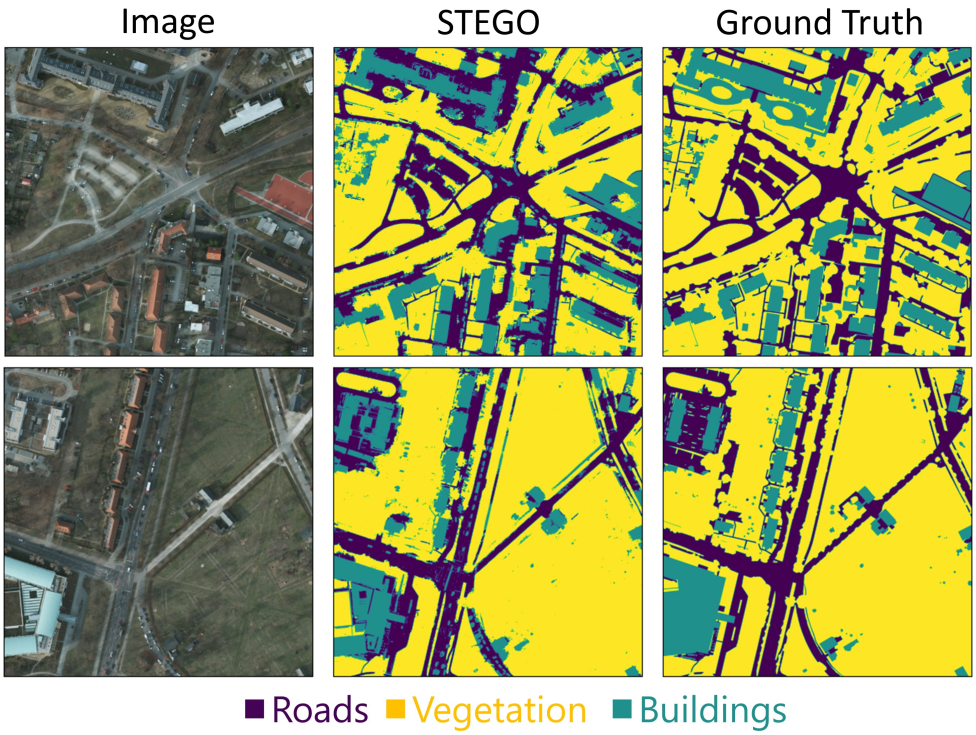


Figure B.1: Qualitative comparison of STEGO segmentation results on the Potsdam-3 segmentation challenge.

B.3 Additional Ablation Study

In addition to the ablation study of Table 4.2, we investigate the effect of each major architectural decision in isolation. We find that in most metrics, removing each architectural component hurts performance.

Table B.2: Additional architecture ablation study on the CocoStuff Dataset (27 Classes).

								Unsupervised		Linear Probe	
Backbone	**0-Clamp**	**5-Crop**	**Pointwise**	**CRF**	**Self-Loss**	**KNN-Loss**	**Rand-Loss**	**Acc.**	**mIoU**	**Acc.**	**mIoU**
ViT-Small	✓	✓	✓	✓	✓	✓	✓	**48.3**	**24.5**	**74.4**	38.3
MoCoV2	✓	✓	✓	✓	✓	✓	✓	43.1	19.6	65.9	26.0
ViT-Small		✓	✓	✓	✓	✓	✓	42.8	10.3	59.3	19.3
ViT-Small	✓		✓	✓	✓	✓	✓	48.0	23.1	73.9	**38.9**
ViT-Small	✓	✓		✓	✓	✓	✓	50.2	22.3	73.7	37.7
ViT-Small	✓	✓	✓		✓	✓	✓	47.7	24.0	72.9	38.4
ViT-Small	✓	✓	✓	✓		✓	✓	43.0	20.2	73.0	36.2
ViT-Small	✓	✓	✓	✓	✓		✓	47.0	22.2	74.0	37.7
ViT-Small	✓	✓	✓	✓	✓	✓		39.8	12.8	65.5	29.9

B.4 Additional Qualitative Results

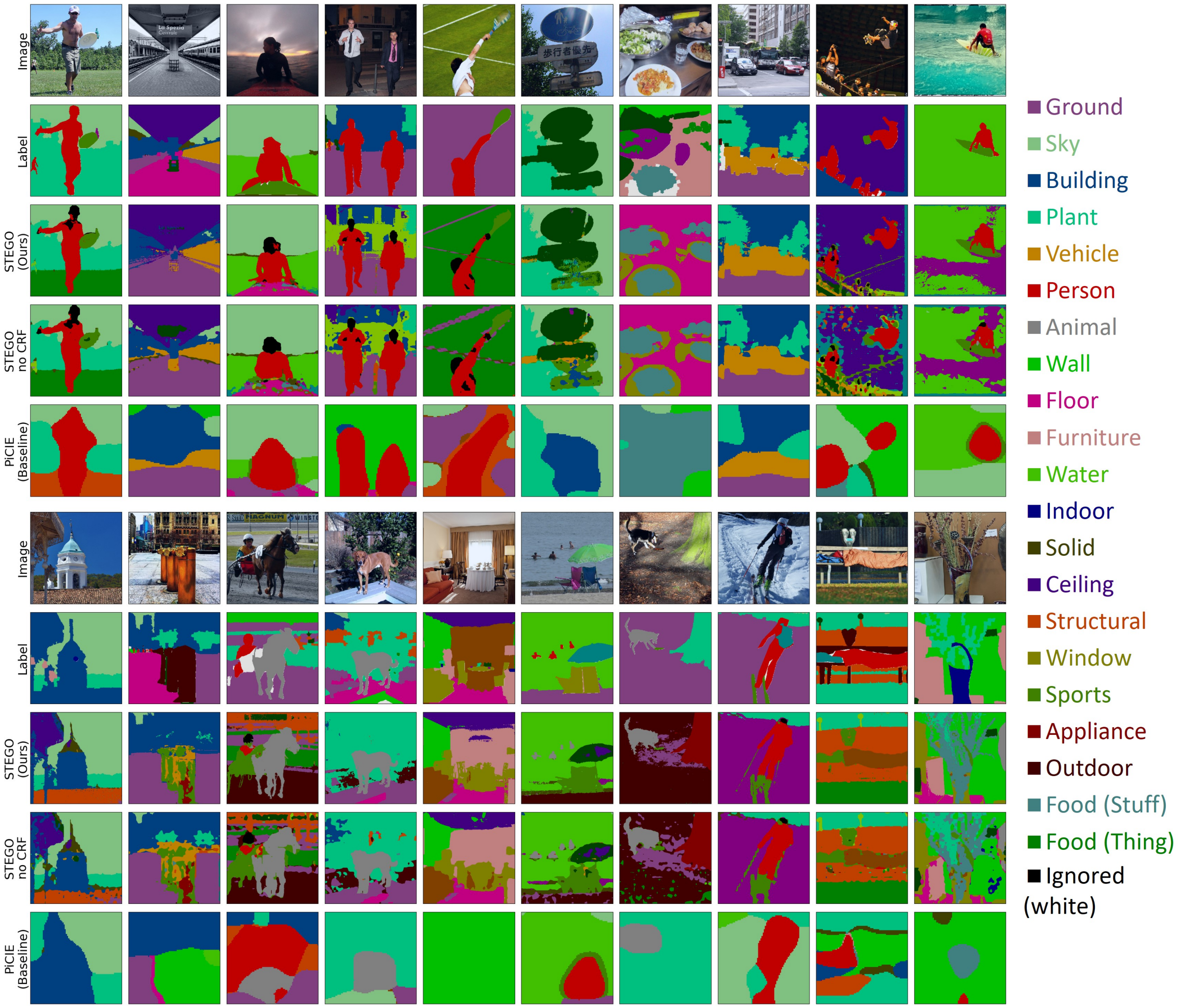


Figure B.2: Additional unsupervised semantic segmentation predictions on the CocoStuff 27 class segmentation challenge using STEGO (Ours) and the prior state of the art, PiCIE. Images are not curated.

B.5 Failure Cases

Unsupervised Segmentation is prone to a variety of issues. We include some of the following to segmentations to demonstrate cases where STEGO breaks down. In the first column of Figure B.3 we can see that STEGO improperly segments ground from trees and backgrounds. In the second column we see that STEGO makes an understandable error and assigns the barn floor to the "outdoor" class and the barn wall to the "building" class. In the third column STEGO misses the boundary between wall and ceiling. The fourth column demonstrates the challenge between food (thing) and food (stuff) characterization. Interestingly PiCIE makes the same type of error both here, and in the barn case. The last column shows an example of STEGO missing a human in the lower left. In this image it is challenging to spot the person, probably because it is grayscale.

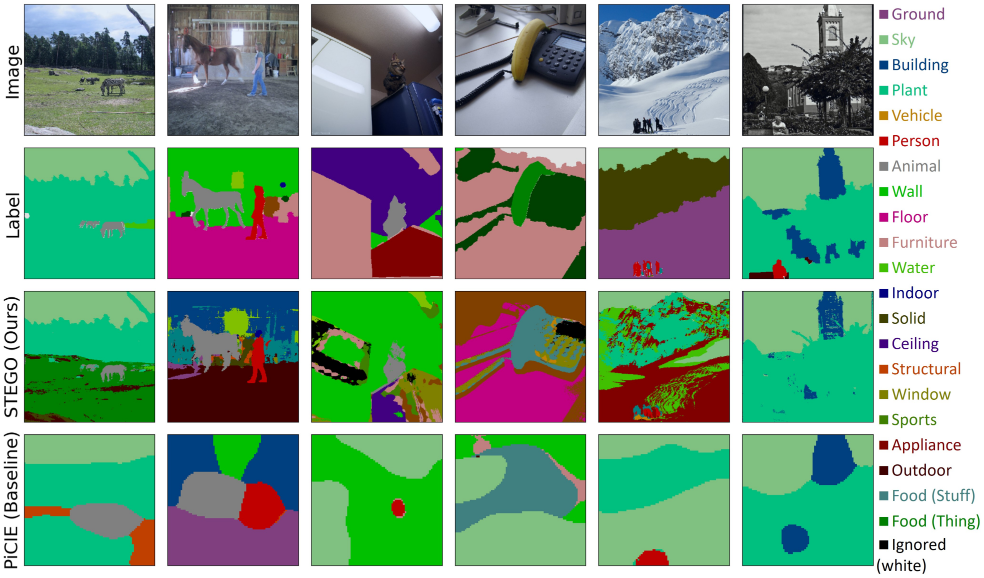


Figure B.3: STEGO failure cases.

B.6 Feature Correspondences Predict STEGO's Errors

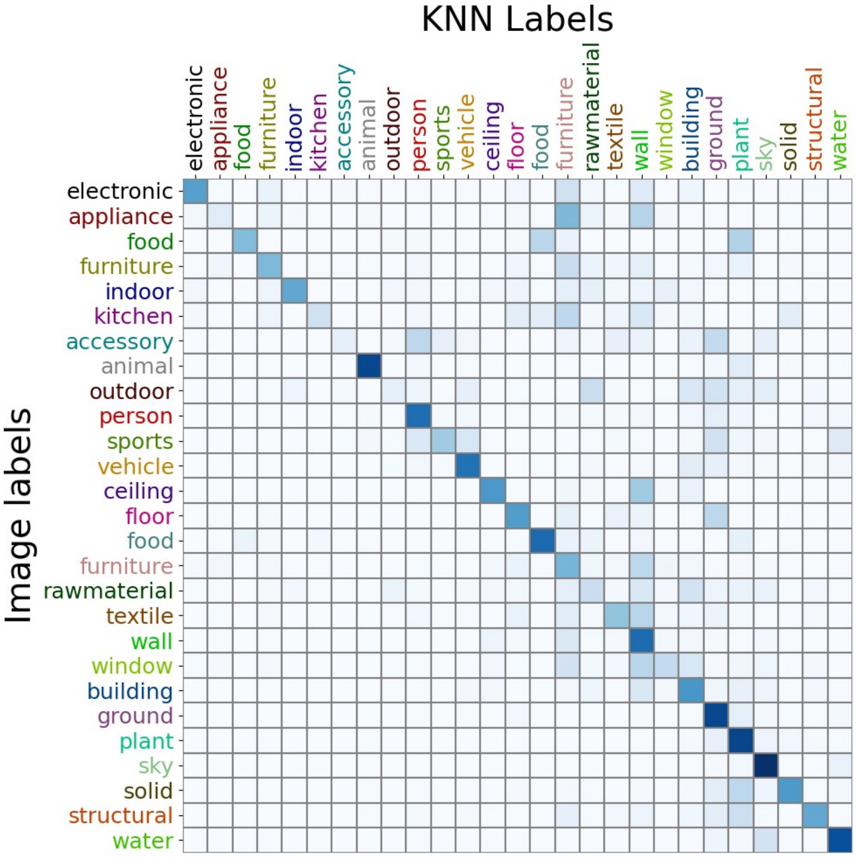


Figure B.4: Normalized matrix of predicted label co-occurrences between an Images and KNNs. This analysis shows where our unsupervised supervisory signal, the DINO feature correspondences, fails to align with the CocoStuff27 label ontology.

Section 4.5.1 demonstrates how unsupervised feature correspondences serve as an excellent proxy for the true label co-occurrence information. In this section we explore how and where DINO's feature correspondences systematically differ from the ground truth labels, and show that these insights allow us to directly predict STEGO's final confusion matrix.

More specifically we consider the setting of Section 4.5.1. Instead of computing precision-recall curves from our feature correspondence scores we can instead threshold these scores, select the strongest couplings between the images, and evaluate whether these couplings are between objects of the same class or objects of different classes. In particular, Figure B.4 shows a confusion matrix capturing how well DINO feature correspondences between images and their K-Nearest Neighbors align with the ground truth label ontology in the CocoStuff27 dataset. We find that that this analysis predicts many of the areas where the final STEGO architecture fails. In particular, we can see that DINO conflates the "Food (things)" and "Food (stuff)" and this error also appears in STEGO's confusion matrix in Figure B.6. Likewise both visualizations show confusion between "appliance" and "furniture", "window" and "wall", and several other common errors.

This analysis demonstrates that many of STEGO's errors originate from the structure of the DINO features used to train STEGO as opposed to other aspects of the architecture. However we note that the question of whether whether this is an issue with the DINO features, or due to ambiguities in the CocoStuff label ontology is still outstanding. Finally we note that this analysis is able to predict the results of a fully-trained STEGO architecture, and could be used as a way to select better backbones without having to training STEGO.

B.7 Higher Resolution Confusion Matrices

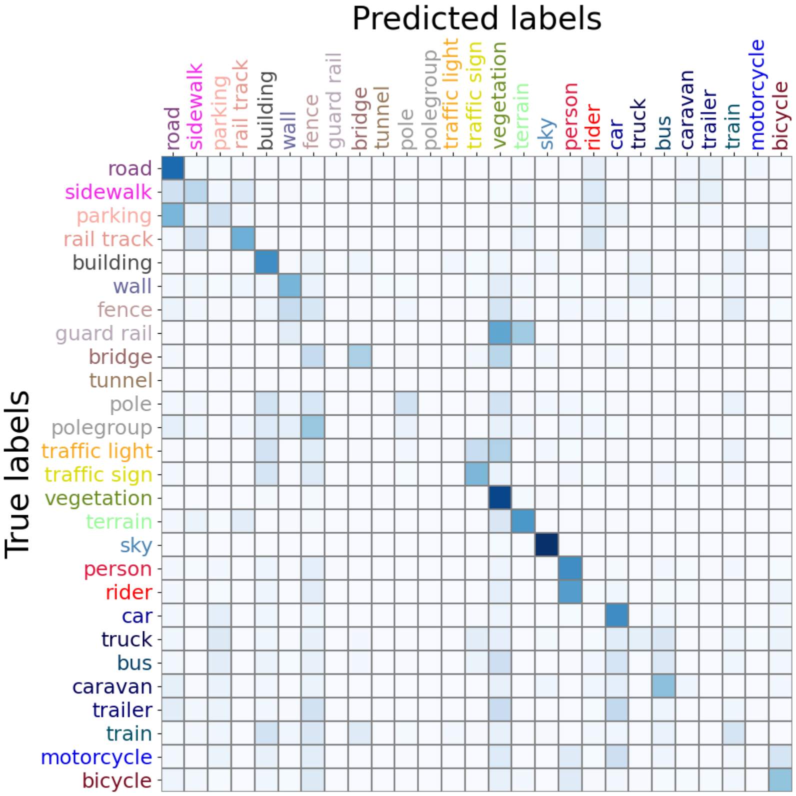


Figure B.5: Confusion Matrix for Cityscapes predictions

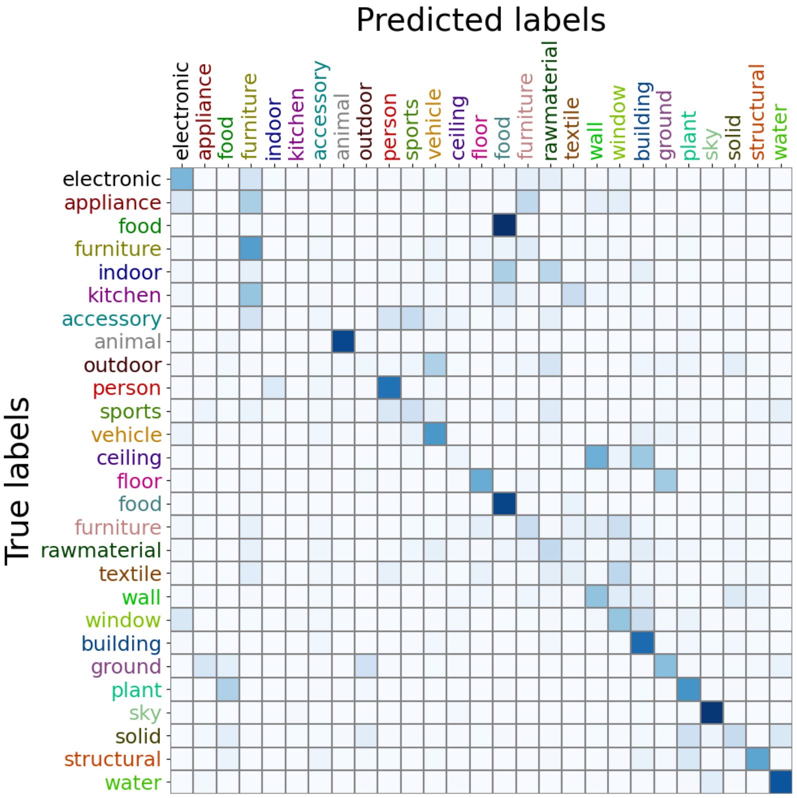


Figure B.6: Confusion Matrix for CocoStuff predictions

B.8 Relationship with Graph Energy Minimization

In section 4.5.4 we briefly mention that STEGO's feature correlation distillation loss defined in Equation 4.4 can be seen as a particular case of Maximum Likelihood (ML) estimation on a undirected graphical model or Ising model. In this section we demonstrate this connection in greater detail using the formalism defined in 4.5.4. In particular, we recall the energy for a Potts model:

$$E(\phi) := \sum_{v_i, v_j \in \mathcal{V}} w(v_i, v_j)\mu(\phi(v_i), \phi(v_j)) \tag{B.1}$$

We then construct the Boltzmann Distribution [130] yields a normalized distribution over the function space Φ:

$$p(\phi|w, \mu) = \frac{\exp(-E(\phi))}{\int_\Phi \exp(-E(\phi'))d\phi'} \tag{B.2}$$

In general, sampling from this probability distribution is difficult because of the often-intractable normalization factor. However, it is easier to compute the maximum likelihood estimate (MLE):

$$\operatorname*{argmax}_{\phi \in \Phi} p(\phi|w, \mu) = \operatorname*{argmax}_{\phi \in \Phi} \frac{1}{Z} \exp(-E(\phi)) \tag{B.3}$$

Where Z is the unknown constant normalization factor. Simplifying the right-hand side yields:

$$\operatorname*{argmax}_{\phi \in \Phi} p(\phi|w, \mu) = \operatorname*{argmin}_{\phi \in \Phi} E(\phi) = \operatorname*{argmin}_{\phi \in \Phi} \sum_{v_i, v_j \in \mathcal{V}} w(v_i, v_j)\mu(\phi(v_i), \phi(v_j)) \tag{B.4}$$

We are now in the position to connect this to the STEGO loss function. First, we take our nodes $\mathcal{V}$ to be the set of all spatial locations across our entire dataset of images. For concreteness we can represent $v \in \mathcal{V}$ by the tuple (n, h, w) where h, w represent height and width n represents the image number. We now let $\phi(v_i)$ be the output of the segmentation head, s_{v_i}, at the image and spatial location v_i. Using cosine distance, $d_{cos}(x, y) = 1 - \frac{x}{|x|}\frac{y}{|y|}$ as the compatibility function, μ, yields the following:

$$= \operatorname*{argmin}_{\mathcal{S}} \sum_{v_i, v_j \in \mathcal{V}} -w(v_i, v_j) \frac{s_{v_i}}{|s_{v_i}|} \frac{s_{v_j}}{|s_{v_j}|} \tag{B.5}$$

Wherte the argmin now ranges over the parameters of the segmentation head $\mathcal{S}$. We can now observe that the sum over all pairs $v_i, v_j \in \mathcal{V}$ can be written as a sum over pairs of images $x, y \in X$ and pairs of spatial locations $(h, w), (i, j)$ where we note that (i, j) in this context refers to the spatial coordinates of image y as in 4.5.1 and not the indices of the vertices.

$$= \operatorname*{argmin}_{\mathcal{S}} \sum_{x, y \in X} \sum_{hwij} -W(x, y)_{hwij} S(x, y)_{hwij} \tag{B.6}$$

Where we define $S(x, y)$ to be the segmentation feature correlation tensor for images x, y as defined in Section 4.5.2. Finally letting $W(x, y)_{hwij} = F_{hwij} - b$ we recover our loss:

$$\underset{\phi \in \Phi}{\operatorname{argmax}}\, p(\phi | w, \mu) = \underset{\mathcal{S}}{\operatorname{argmin}} \sum_{x,y \in X} \mathcal{L}_{simple-corr}(x, y, b) \tag{B.7}$$

Finally we note that in practice we approximate the minimization using minibatch SGD, and our inclusion of KNN and Self-correspondence distillation changes the weight function w, but does not change its functional form.

Switching to the ML formulation of this problem allows us to solve this optimization for ϕ by gradient descent on the parameters of the segmentation head, $\mathcal{S}$, and makes this computationally tractable. For large image datasets that can contain millions of high-resolution images, the induced graph can contain billions of image locations. Other graph embedding and clustering approaches such as Spectral methods require solving for eigenvalues of the graph Laplacian, which can take $O(|\mathcal{V}|^3)$ time [376]. More recent attempts to accelerate Spectral clustering such as [376] and [377] further assume a "Nonparametric" structure on the function ϕ, where a separate cluster assignment is learned for each vertex. This assumption of a "nonparametric" function ϕ can be undesirable as one cannot cluster or embed new data without recomputing the entire clustering. In contrast, STEGO's backbone and segmentation head act as a parametric form for the function ϕ allowing the approach to output predictions for novel images.

B.9 Continuous, Unsupervised, and Mini-batch CRF

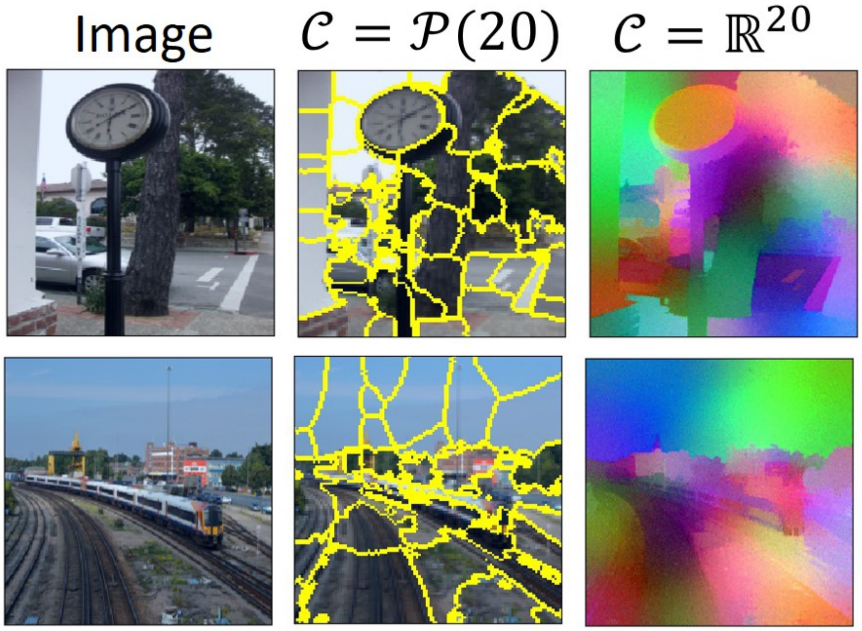


Image
$\mathcal{C} = \mathcal{P}(20)$
$\mathcal{C} = \mathbb{R}^{20}$

Figure B.7: Unsupervised CRF solutions for discrete (middle) and continuous (right) code spaces. In the discrete case we mark the boundaries between classes, in the continuous case we visualize the top 3 dimensions of the code space.

Fully connected Gaussian Conditional Random Fields (CRFs) [378] are an extremely popular addition to semantic segmentation architectures. The CRF has the ability to improve initial predictions of locations, and can "sharpen" predictions to make them consistent with edges and areas with consistent color in the original image. CRF post-processing for refining supervised and weakly supervised semantic segmentation predictions is ubiquitous in the literature [91, 180, 378–380]. Recently, new connections between CRF message passing and convolutional networks have allowed CRFs to be embedded into existing models [381, 382] and trained jointly for better performance. By connecting the STEGO correspondence distillation loss to the energy of an undirected model on image pixels we can use the same minibatch MLE strategy to estimate other similar graphical models. For example, in the fully connected Gaussian edge potential CRF, one forms a pairwise potential function potential function for the pixels of a single image:

$$w_{crf}(v_i, v_j) = a \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) + b \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right) \tag{B.8}$$

Where p_i represent the pixel coordinates associated with node v_i and I_i represents pixel colors associated with node v_i. The parameters $a, b, \theta_\alpha, \theta_\beta, \theta_\gamma$ are hyperparameters and control the behavior of the model. These parameters balance the effect of long- and short-range color similarities against smoothness. The CRF directly learns a pixel-wise array of probabilistic class assignments over k labels corresponding to the probability simplex code space $\mathcal{C} = \mathcal{P}(l)$ and a non-parametric clustering function f. For a compatibility function μ the CRF chooses the Potts Model [128]: $\mu_{potts}(\phi(v_i), \phi(v_j)) := \mathbb{P}(\phi(v_i) \neq \phi(v_j))$.

With this setting of the weights and compatibility function, we directly recover the binary potentials of the fully connected Gaussian edge potential CRF [118]. We can also add the unary potentials which are often the outputs of another model. However, for our analysis we explore the case without unary potentials which yields an "unsupervised" variant of the CRF. However, without external unary potential terms, the strictly positive similarity kernel encourages the maximum likelihood estimator (MLE) of the graph to be the constant function. To rectify this, we can add small negative constant, $-b$, to the weight tensor to push unrelated pixels apart. This negative force is the direct analogue of the negative pressure hyper-parameter in STEGO and can be interpreted through the lens of negative sampling [383]. This negative shift also appears in the word2vec and graph2vec embedding techniques [384, 385]. Our shifted CRF potential encourages natural clusters to form that respect the structure of the potentials that capture similarities in pixel colors and locations. In the discrete case, solutions to this equation resemble superpixel algorithms such as SLIC [386]. Additionally lifting this to the continuous code space and provide a natural continuous generalization of superpixels and seems to avoid challenging local minima. We illustrate these solutions to just the unsupervised CRF potential in Figure B.7. Finally, we note that the second term of Equation B.8, referred to as the smoothness kernel, matches IIC's notion of local class consistency. However, we found that adding these CRF terms to the self-correspondence loss of STEGO did not improve performance.

B.10 Implementation Details

Model STEGO uses the "ViT-Base" architecture of DINO pre-trained on ImageNet. This backbone was trained using self-supervision without access to ground-truth labels. We use the "teacher" weights when creating our backbone. We take the final layer of spatially varying features and apply a small amount ($p = 0.1$) of channel-wise dropout [212] before using them throughout the architecture during training. Our segmentation head consists of a linear network and a two-layer ReLU MLP added together and outputs a 70 dimensional vector. We use the Adam optimizer [387] with a learning rate of 0.0005 and a batch size of 32. To make our losses resolution independent we sample 121 random spatial locations in the source and target implementations and use grid sampling [388] to sample features from the backbone and segmentation heads. Our cluster probe is trained alongside the STEGO architecture using a minibatch k-means loss where closeness is measured by cosine distance. Cluster and linear probes are trained with separate Adam optimizers using a learning rate of .005

Datasets We use the training and validation sets of Cocostuff described first in [93] and used throughout the literature including in [85]. We note that the validation set used in [93] is a subset of the full CocoStuff validation set and we use this validation subset to be consistent with prior benchmarks. We note that using the full validation set does not change results significantly. When five-cropping images we use a target size of $(.5h, .5w)$ for each crop where h, w are the original image height and width. Training images are then scaled to have minor axis equal to 224 and are then center cropped to $(224, 224)$, validation images are first scaled to 320 then are center cropped to $(320, 320)$. All image resizing uses bilinear interpolation and resizing of target tensors for evaluation uses nearest neighbor interpolation.

CRF We use PyDenseCRF [118] with 10 iterations with parameters $a = 4, b = 3, \theta_\alpha = 67, \theta_\beta = 3, \theta_\gamma = 1$ as written in Section B.9.

Compute All experiments use PyTorch [131] v1.7 pre-trained models, on an Ubuntu 16.04 Azure NV24 Virtual Machine with Python 3.6. Experiments use PyTorch Lightning for distributed and multi-gpu training when necessary [389].

Table B.3: Hyperparameters used in STEGO

Parameter	**Cityscapes**	**CocoStuff**
λ_{rand}	0.91	0.15
λ_{knn}	0.58	1.00
λ_{self}	1.00	0.10
b_{rand}	0.31	1.00
b_{knn}	0.18	0.20
b_{self}	0.46	0.12

Hyperparameters We use the following hyperparameters for our results in Tables 4.1 and 4.3:

B.11 A Heuristic for Setting Hyper-parameters

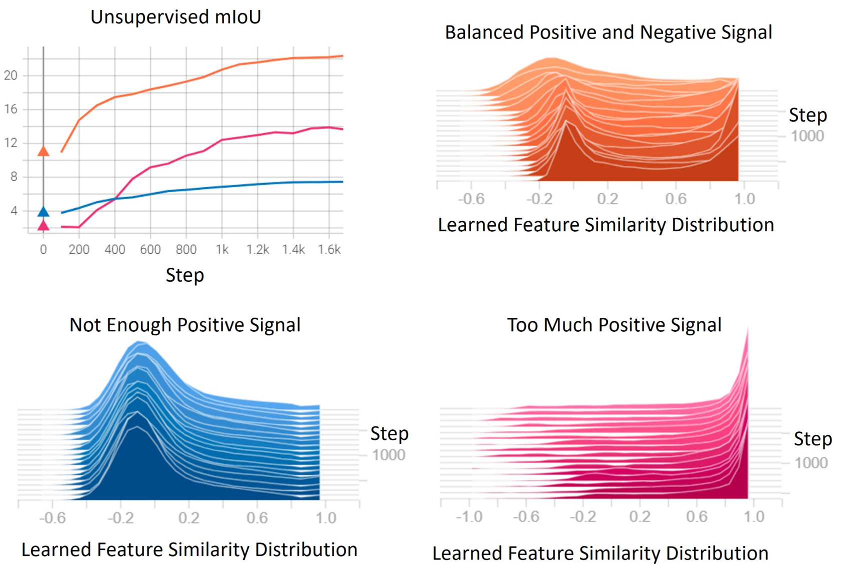


Figure B.8: Distributions of feature correspondences between an image and itself across three different hyper-parameter settings. The orange curve and distribution shows a proper balance between attractive and repulsive forces allowing some pairs features to cluster together (the peak at 1) and other pairs of features to orthogonalize (the peak at 0)

Setting hyperparameters without cross-validation on ground truth data can be difficult and this is an outstanding challenges with the STEGO architecture that we hope can be solved in future work. Nevertheless we have identified some key intuition to guide manual hyperparameter tuning. More specifically, we find that the most important factor affecting performance is the balance of positive and negative forces. Too much negative feedback and vectors will all push apart and clusters will not form well, too much positive feedback and the system will tend towards a small number of clusters. To debug this balance, we found it useful to visualize the distribution of feature correspondence similarities as a function of training step as shown in Figure B.8. A balanced system (Orange distribution) will tend towards a bi-modal distribution with peaks at alignment 1 or orthogonality at 0. This bi-modal structure is indicative that there is some clustering within images, but that not everything is assigned to the same cluster. Pink and blue distributions show too much positive and negative signal respectively. We find that given a reasonable balance of the λ's, this balance can be achieved by tuning the bs to achieve the desired balance.

B.12 A note on 5-Crop Nearest Neighbors

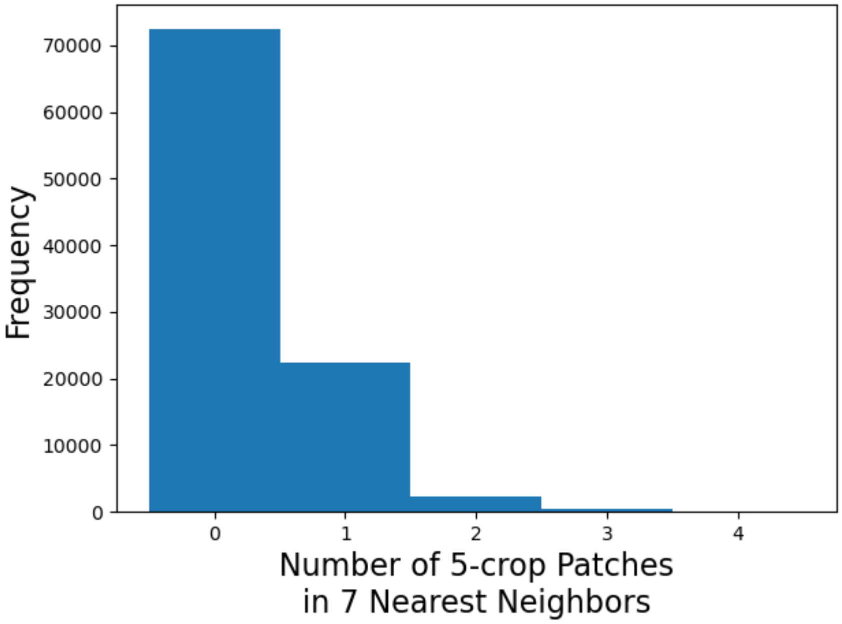


Figure B.9: Number of patches from the same image found within each patch's 7 nearest neighbors

We found that pre-processing the dataset by 5-cropping images was a simple and effective way to improve the spatial resolution of STEGO and the quality of K-Nearest Neighbors. We consider each resulting 5-crop as a separate image when computing KNNs and patches from the same image are valid KNNs. Figure B.9 shows the distribution of these self-matches for the CocoStuff dataset. We note that the majority of patches do not have any nearest neighbors from the same image.

Appendix C

Appendix for Chapter 5

C.1 Website, Video, and Code

We provide additional details and a short video explaining FeatUp at aka.ms/featup. Additionally, we provide our code at: https://tinyurl.com/28h3yppa

C.2 Strided baseline implementation

For the DINO and ViT backbones, we extract patches with a stride of $\frac{16}{\text{upsample factor}}$ to produce a higher density of feature vectors and thus increase feature resolution. We point out that the upsampling factor is limited with this method (as the stride is lower bounded by 1), so this approach can only upsample up to 16x for ViT-S/16. Practically however, these maximum upsampling factors are impractical as they require far more memory than current GPUs provide (see Figure C.10).

C.3 Comparison to Image-Upsampling Methods

A variety of methods have been proposed for image super-resolution. Among the learning-based approaches, deep image prior (DIP) [178] has been used succesfully for enhancing images without additional training data. Figure C.1 shows that DIP poorly upsamples features, introducing artifacts and “blob" patterns in the features and downstream outputs. [176] introduced Zero-Shot Super-Resolution, a method that learns an image-specific CNN at test time without additional training data. Additionally, images can be represented as Local Implicit Image Functions (LIIF) [177] which can be queried at arbitrary resolution. While similar to FeatUp's implicit network, LIIF trained to continuously represent a feature map does not produce sharp outputs like FeatUp (Figure C.1) Despite these methods' successes in the image super-resolution problem space, they are not equipped to upsample high-dimensional features.

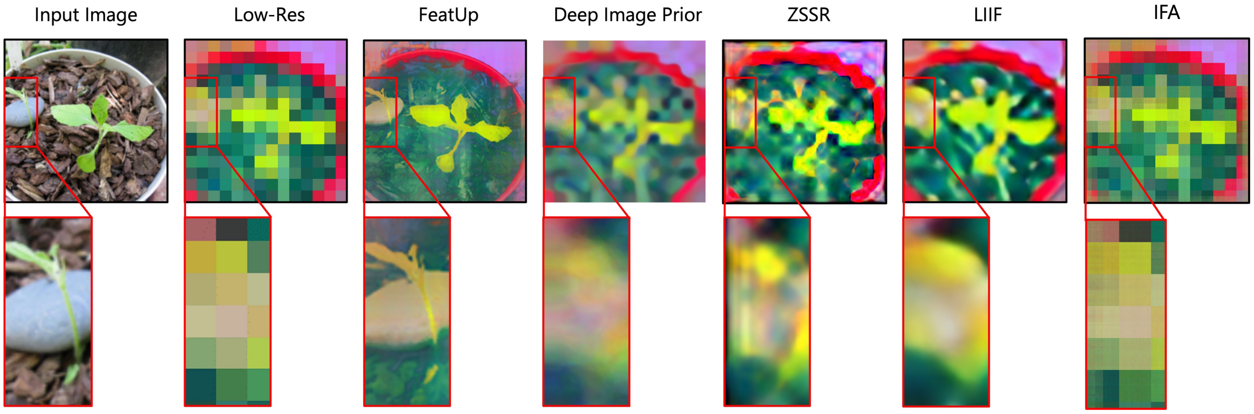


Figure C.1: Comparison of image super-resolution methods using Deep Image Prior, Zero-Shot Super-Resolution (ZSSR), and Local Implicit Image Function (LIIF). We also include a visualization on Implicit Feature Alignment (IFA). As shown in the whole feature map and zoomed-in section, thse image upsampling methods do not effectively upsample the low-resolution and high-dimensional feature maps by the large upsampling factors that we are able to handle.

C.4 Ablation Studies

We show the effects of each design decision for FeatUp in Figure C.2. Our upsampler blurs ResNet features without the uncertainty loss, possibly because it cannot ignore certain nonlinear artifacts or resolve the large pooling window present in ResNet-50. The magnitude regularizer provides smoothing and regularization benefits. Our choice to include Fourier color features dramatically improves resolution and high-frequency details. Finally, the attention downsampler helps the system avoid odd edge and halo effects by learning kernels more focused on salient parts of the signal. Using an explicit buffer of features instead of an implicit network yields significant artifacts, though we note that the artifacts are significantly less dramatic if the simple downsampler is also used.

We also provide an ablation study of the total variation and magnitude regularizers in Figure C.4. Our regularizer is fairly robust to different settings as shown by the 2x multiplication for both terms in the 3rd column. However, there still exists an optimal λ range that provide important smoothing properties; larger values can interfere with the main reconstruction objective as shown in the final column.

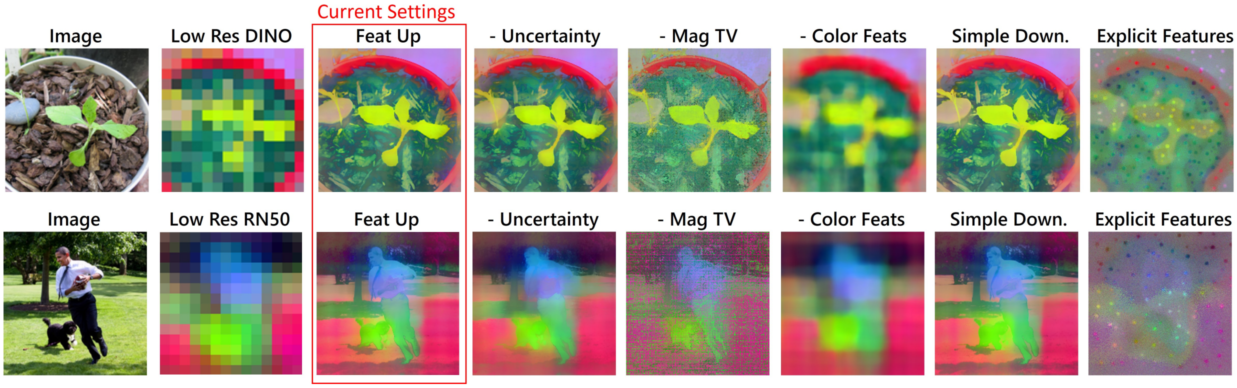


Figure C.2: Qualitative ablation study across both DINO and Resnet50 Backbones. The biggest improvements arise from the implicit featurizer, color features, and the magnitude TV regularizer.

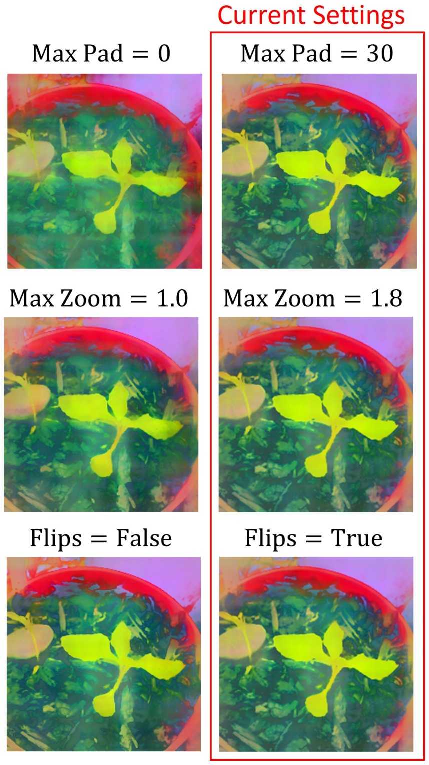


Figure C.3: Ablation of FeatUp's training hyper-parameters. We are robust to a range of jitter values, though features degrade with large changes in max pad.

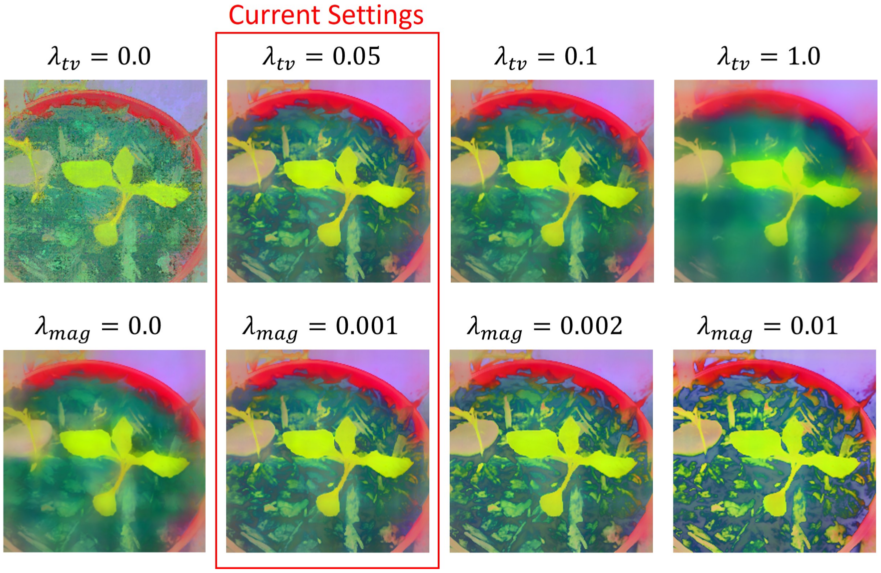


Figure C.4: Qualitative ablation study of the TV and magnitude regularizers. FeatUp is fairly robust to the settng of these parameters.

To further justify our design decisions in the context of an end-to-end trained architecture, we evaluate JBU with the Segformer [218] decoder by 1) removing the MLP (denoted as MLP in Equation 5.6) on the guidance signal, 2) removing the temperature-weighted softmax and replacing it with Euclidean distance between the central feature and its neighborhood, and 3) removing the softmax and replacing it with cosine distance. Each ablation degrades segmentation performance, with the MLP exclusion being the most detrimental.

	FeatUp (JBU)			
	Original	- MLP	- Softmax + Euclidean Dist.	- Softmax + Cosine Dist.
mIoU	44.2	42.9	43.8	43.7
mAcc	55.8	54.7	54.5	55.3
aAcc	80.7	79.4	80.0	80.4

Table C.1: Semantic segmentation performance with the Segformer architecture trained on the ADE20k train set and evaluated on the val set. Ablated FeatUp (JBU) replaces the original feature upsampling in the Segformer decoder.

	CAM Score		Semantic Seg.		Depth Estimation	
Ablation	A.D. ↓	A.I. ↑	Acc. ↑	mIoU ↑	RMSE ↓	δ >1.25 ↑
Original	**9.83**	**5.24**	**68.77**	**43.41**	**1.09**	**0.938**
- MLP	10.04	5.10	68.12	42.99	1.14	0.917
- Softmax + Euclidean	9.98	5.19	68.68	43.16	1.10	0.928
- Softmax + Cosine	9.97	5.21	68.49	43.15	1.12	0.924

Table C.2: FeatUp (JBU) performance with ablated architectural components: removing the MLP, replacing softmax with a gaussian kernel w.r.t. Euclidean or cosine distance. Across all metrics, each ablation degrades performance.

			CAM Score		Semantic Seg.		Depth Estimation	
Attn DS.	O.D.	TV Reg.	A.D. ↓	A.I. ↑	Acc. ↑	mIoU ↑	RMSE ↓	δ >1.25 ↑
✓	✓	✓	**8.84**	**5.60**	**71.58**	**47.37**	**1.04**	**0.927**
✗	✓	✓	9.07	5.06	70.95	46.79	1.11	0.916
✓	✗	✓	8.91	5.55	71.26	46.89	1.08	0.920
✓	✓	✗	9.10	5.00	68.06	44.36	1.11	0.913

Table C.3: Ablation study for implicit FeatUp features with varied downsampler (attention = ✓, simple = ✗), outlier detection, λ_{TV} (0.05 = ✓, 0.0 = ✗).

C.5 Visualizing Additional PCA Components

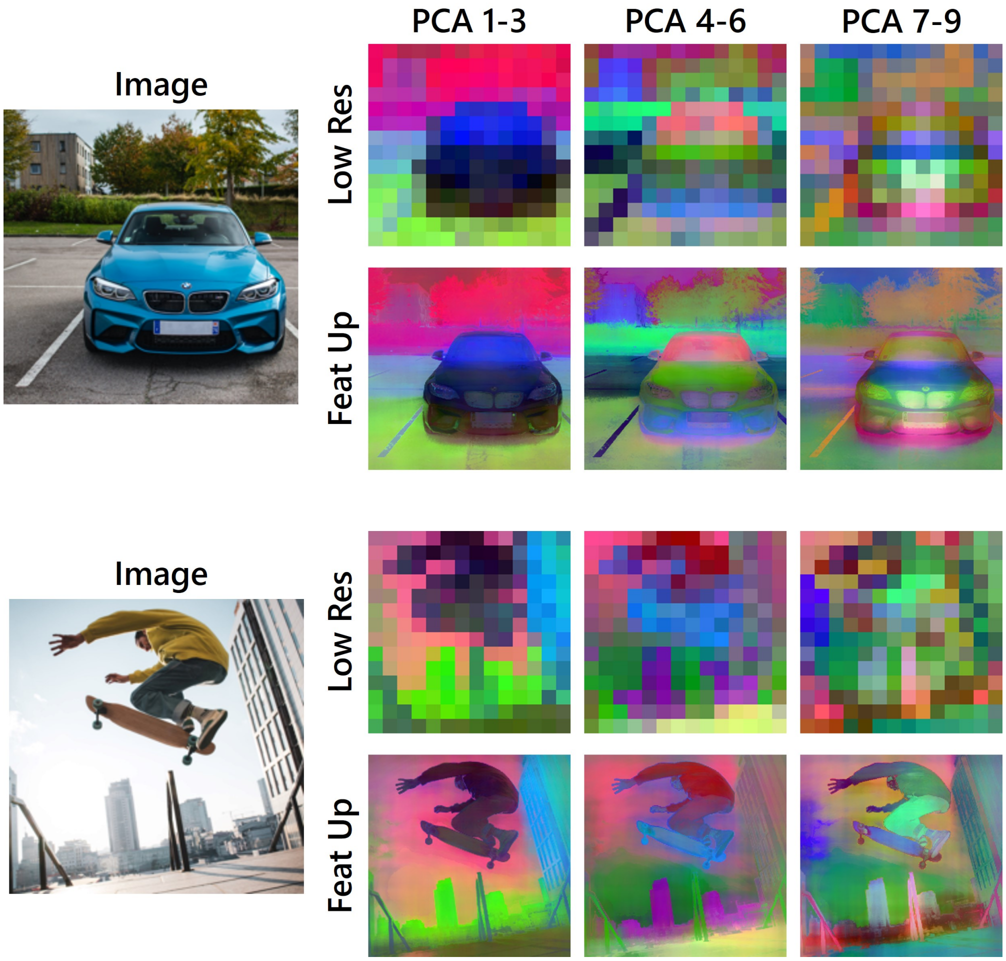


Figure C.5: Visualizing higher PCA components with FeatUp. FeatUp upsamples entire feature maps, so their higher-order principal components also remain in the same space as the original features and are upsampled precisely. Higher components tend to separate more fine-grained object categories like the skater from the skateboard, and the trees from the background, and the clouds from the sky. Note that each subobject's features are upsampled precisely to the object it represents.

C.6 Saliency Map Details

Downsampling in FeatUp is analogous to ray-marching in NeRF, which approximates the physics of image formation. FeatUp's downsampler approximates a network's process of pooling information into features. As shown in Figure 6, most networks preserve the rough location of objects in their features (the objects just appear downsampled and blurred). This observation leads us to use blur/pooling operators.

The simplest of these is average pooling, but we can do better by generalizing this operation to a learned blur/pooling kernel so the downsampler can better match a network's receptive field size. To map back to NeRF, this is like adding learned camera lens distortion parameters to the ray-marcher so NeRF can better fit the data.

As shown in Figure 5.6 and described in Section 3.1, even a learned blur/pooling kernel cannot capture dynamic receptive fields or object salience. For example if a small amount of an important object is in a transformer's patch, the whole feature changes. We capture effects like this by making the learned pool/blur kernel dependent on image content using a 1x1 conv (we don't need anything bigger than this layer). This generalizes the previously-described learned blur/pool and allows the downsampler to adaptively pool based on image content. Figure C.6 shows that the salience network focuses on certain attributes (e.g. object boundaries, some important small objects). We also note that many common pooling strategies such as average pooling or nearest/bilinear/bicubic resizing are special cases of our learnable attention pooling strategy.

C.7 Visualizing Downsampler Salience and Kernels

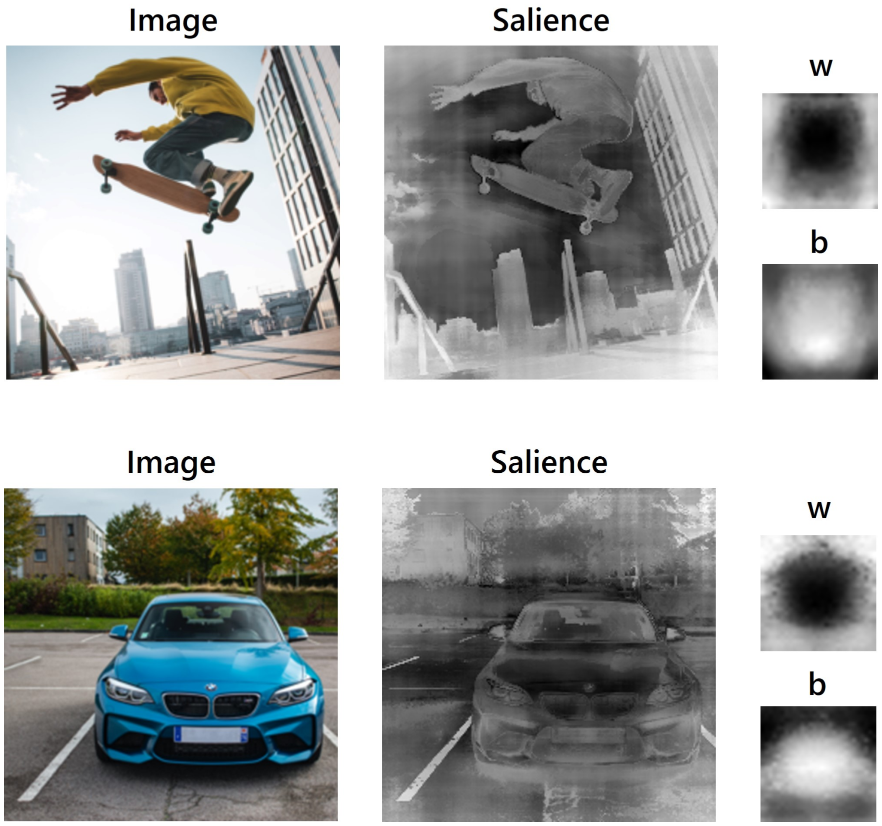


Figure C.6: Visualization of downsampler salience and weight and bias kernels for two images. Note how fine-grained objects have higher salience and regions around important objects (like the sky between the hands and the skateboard) have lower salience. This allows the network to capture nonlinear behavior where embeddings from salient regions dominate the embeddings of other regions.

C.8 Visualizing Predicted Uncertainty

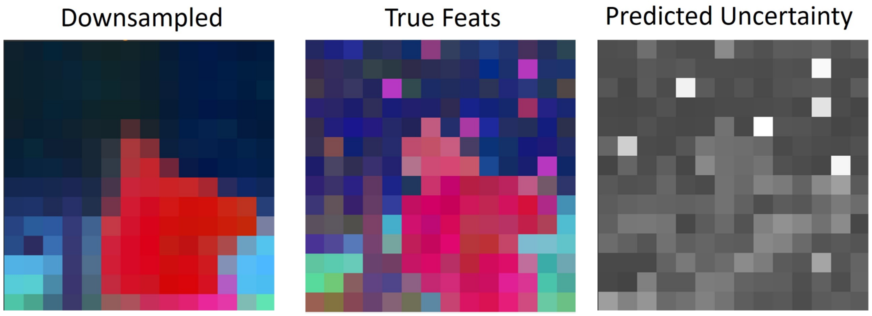


Figure C.7: An example predicted uncertainty map for a set of ViT features. White areas have higher uncertainty. In this figure, we can see that nonlinear artifacts like the spurious pink tokens are marked with high uncertainty as they change location depending on the given evaluation. These tokens might serve some other role in the network, such as class-token-like information aggregation. We do not see these types of effects in DINO or convolutional networks.

C.9 Improving Image Retrieval for Small Objects

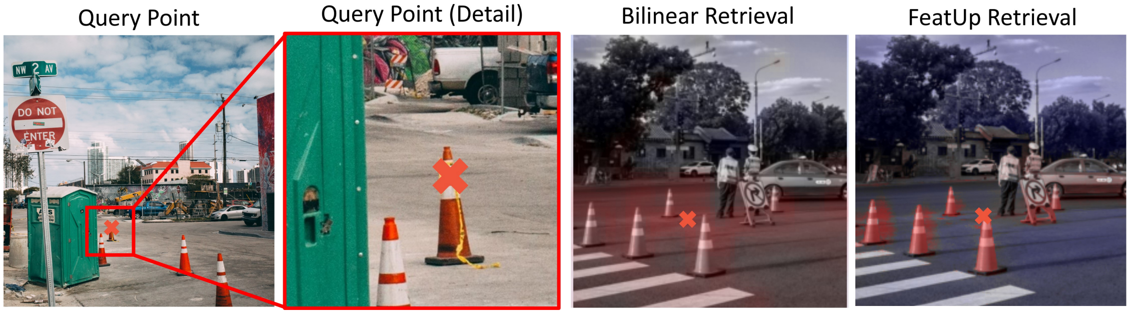


Figure C.8: High-resolution FeatUp features can be used to improve the retrieval of small objects and cluttered scenes. A query image (Left) is featurized with DINO and the region marked with a red × is used as a query point. We show the detailed placement of this query point in the second image from the left. In the two images on the right, we show the closest matching point in the target image (red ×) and we also visualize the similarity heatmap (red means similarity, blue means dissimilarity). The second image from the right depicts the matching point and heatmap when using bilinear feature interpolation on the image and target. The image on the far right shows the results after upsampling with FeatUp prior to computing the retrieval. Because the scene is cluttered, bilinear interpolation blurs object features together and the resulting query vector attends over both the ground and the traffic cones. FeatUp's features better align with objects allowing only the traffic cones to be retrieved.

C.10 Linear Probe details

In both linear probe tasks, one probe was trained on low-resolution (14x14) features from the COCO training set, and frozen for validation across all methods. FeatUp's performance improvements on this repurposed linear probe show that our methods increase resolution without compromising the original feature space. We highlight that these results are *not meant to improve state-of-the-art performance* on segmentation and depth estimation; they are meant to showcase *feature quality* across upsamplers. Because prediction for both tasks is done with a frozen backbone and a single trainable linear probe, the segmentation and depth maps are not meant as a direct application.

C.11 Average Drop and Average Increase Details

Average Drop is expressed as $\sum_{i=1}^{N} \frac{max(0, Y_i^c - O_i^c)}{Y_i^c} \cdot 100$, where Y_c^i is the classifier's softmax output (i.e. confidence) on sample i for class c, and O_i^c is the classifier's softmax output on the CAM-masked sample i for class c. We generate O_i^c by keeping the top 50% of CAM values (and Gaussian blurring the remaining 50% of values with less explainability power). Though we generally expect classifiers to drop in confidence because even masking out less-salient pixels can remove important image context, a high-quality CAM will target the explainable regions of an image more precisely and thus maintain a higher confidence. In the reverse direction, we measure the Average Increase to capture the instances where CAM-masked inputs increase model confidence. Specifically, we define Average Increase as $\sum_{i=1}^{N} \frac{\mathbb{1}_{Y_i^c < O_i^c}}{N} \cdot 100$ where $\mathbb{1}_{Y_i^c < O_i^c}$ is an indicator function equal to 1 when $Y_i^c < O_i^c$ - that is, when model confidence increases upon classifying a CAM-masked image.

Similar to the RelevanceCAM evaluation in [154], we randomly select 2000 images from the ImageNet validation set (limited to images where the label and model prediction match) to measure A.D. and A.I. on.

C.12 Performance Benchmarking

See Table C.4 for performance benchmarking of our adaptive convolution CUDA kernel used in FeatUp (JBU).

Shape (B, H, W, C, F)	Method	Forward (ms)	Backward (ms)	Peak Mem (Mb)
1 × 14 × 14 × 2048 × 5	Ours	**0.15**	**1.05**	**6.24**
	TorchScript	2455	69367	12.8
	Unfold	3.30	2.81	119.
1 × 512 × 512 × 3 × 5	Ours	**0.55**	**2.10**	**10.2**
	TorchScript	147.	520.	24.3
	Unfold	3.47	4.85	231.
16 × 32 × 32 × 2048 × 5	Ours	**8.43**	**90.8**	**372.**
	Unfold	118.	218.	6628.
32 × 512 × 512 × 3 × 5	Ours	**17.7**	114.	**326.**
	Unfold	36.0	**104.**	4901.
64 × 14 × 14 × 2048 × 5	Ours	**6.12**	**61.1**	**400.**
	Unfold	57.5	170.	5174.
64 × 224 × 224 × 3 × 5	Ours	**6.27**	36.1	**128.**
	Unfold	16.7	**27.4**	1878.
64 × 64 × 64 × 16 × 5	Ours	**1.06**	**8.99**	**44.5**
	Unfold	7.18	14.5	822.
64 × 64 × 64 × 16 × 7	Ours	**2.00**	**8.36**	**52.6**
	Unfold	10.8	25.6	1596.

Table C.4: Comparing the performance of our CUDA JBU kernel with with implementations based on PyTorch's `Unfold` operation and TorchScript. Our implementation dramatically reduces memory overhead and increases inference speed. Code for this operation is available in the provided link.

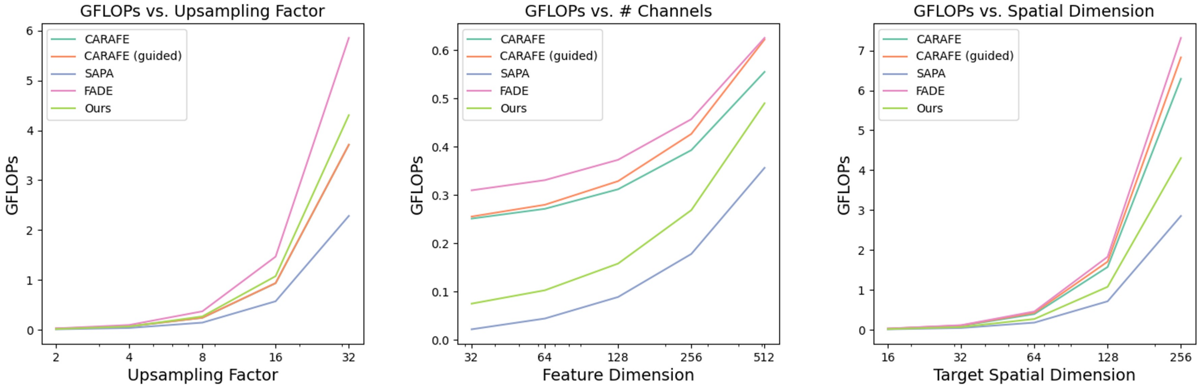


Figure C.9: We evaluate how floating point operations scale with various factors. In varying the upsampling factor, feature dimension, and target spatial dimension, FeatUp (JBU) remains competitive in GFLOP usage. For each experiment, the attributes not studied are kept constant (upsampling factor = 2, feature dimension = 256, starting spatial dimension = 8x8).

We analyze peak memory usage and inference time for various upsampling methods. Specifically, we upsample ViT features from a $(1 \times 3 \times 224 \times 224)$ image (i.e. low-resolution feature dimensions of $(1 \times 384 \times 14 \times 14)$) by factors of 2, 4, 8, and 16. Figure C.10 shows that FeatUp (JBU)'s peak memory closely follows resize-conv and SAPA baselines and outperforms CARAFE. Additionally, FeatUp is as fast as baselines yet outperforms baselines in all our quantitative evaluations. We note that strided and large image baselines become computationally infeasible after 8× upsampling, even using a batch size of 1.

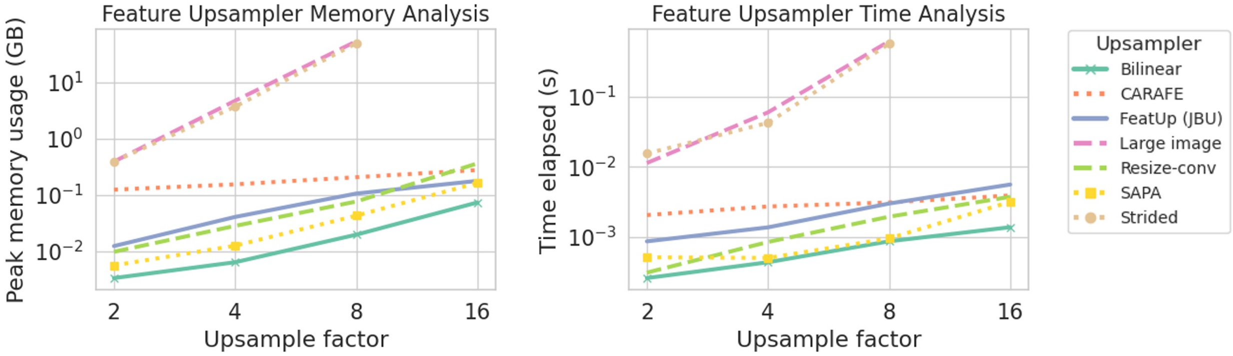


Figure C.10: Analysis of peak memory usage (left) and inference time (right) for various forward-pass upsamplers. FeatUp (JBU) is competitive with SAPA and resize-conv across upsampling factors and is more efficient than CARAFE for smaller factors. The large image and strided approaches become infeasible at large upsampling factors we only show metrics for these methods up to 8× upsampling.

C.13 Additional Qualitative Results

We provide additional CAM visualizations with supervised ViT features on the ImageNet val set in Figure C.11. As in the main chapter, we upsample features from 14x14 to 224x224 output before extracting CAMs (except for the "Low-Res" column, where the features are kept as-is). Both FeatUp (JBU)'s edge-preserving bilateral filters and the FeatUp (Implicit)'s feature representation allow resulting CAMs to highlight salient regions more accurately. Our CAMs combine the semantic advantages of low-resolution features and the spatial advantages of large images, producing refined versions of the original CAMs without discontinuous patches present in the other upsampling schemes.

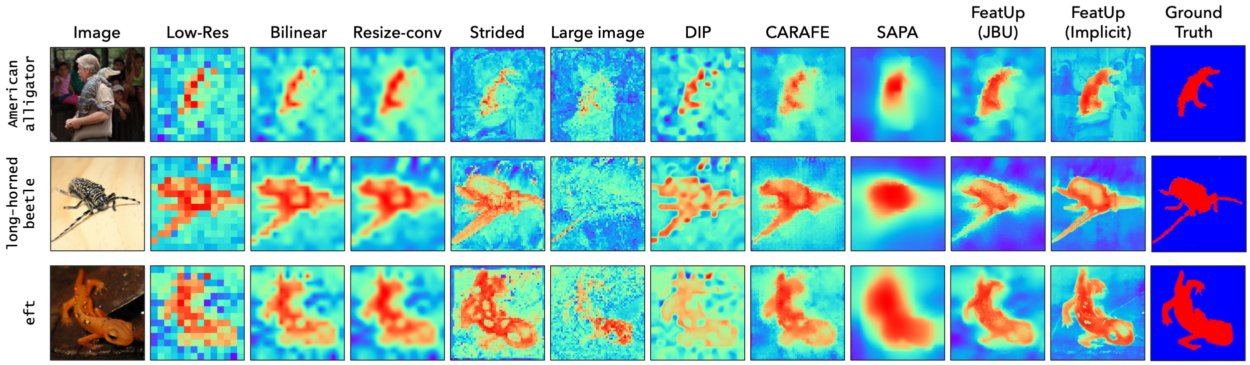


Figure C.11: CAMs on the ImageNet validation set from a supervised ViT backbone and linear probe classifier. Both FeatUp variants produce features that are more precise with respect to the input image, allowing downstream CAMs to better align with object boundaries.

See Figure C.12 for examples of linear probe transfer learning for semantic segmentation on the COCO-Stuff dataset. The 14x14 features output from a ViT backbone are upsampled with the following methods to achieve 224x224 resolution. Then, a linear probe is trained on the low-resolution features and frozen for evaluation on COCO-Stuff semantic class labels. Our methods recover more cohesive labels of objects and backgrounds.

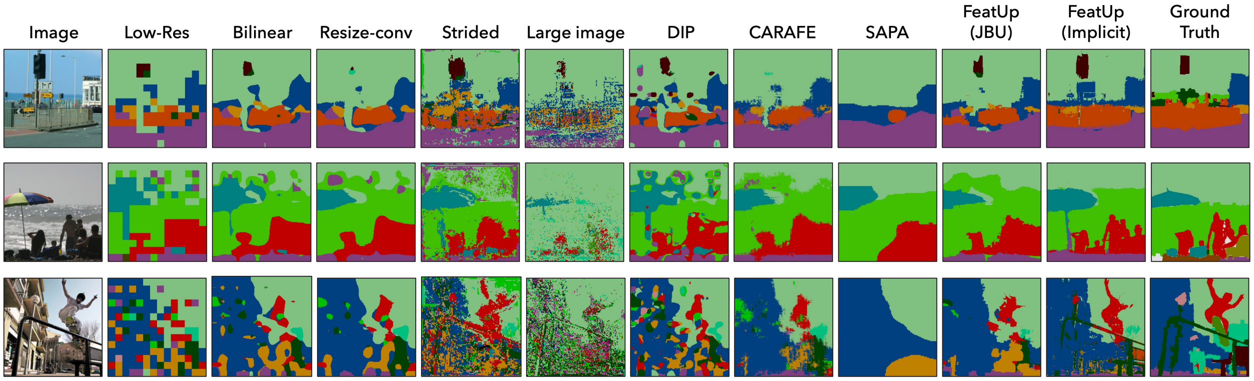


Figure C.12: Examples of linear probe transfer learning for semantic segmentation on the COCO-Stuff dataset. Our methods more closely resemble ground-truth segmentation and smooth many of the artifacts present in the low-resolution features. Additionally, FeatUp (Implicit) recovers thin structures like the umbrella pole not even present in the ground truth despite being semantically correct.

Figure C.13 provides additional examples of linear probe transfer learning for depth estimation. The 14x14 features output from a ViT backbone are upsampled to achieve 224x224 resolution. Then, a linear probe is trained *directly on the features* to predict depth while supervised with a small MiDaS network. Our results show that both FeatUp variants result in high-quality features capable of transfer learning.

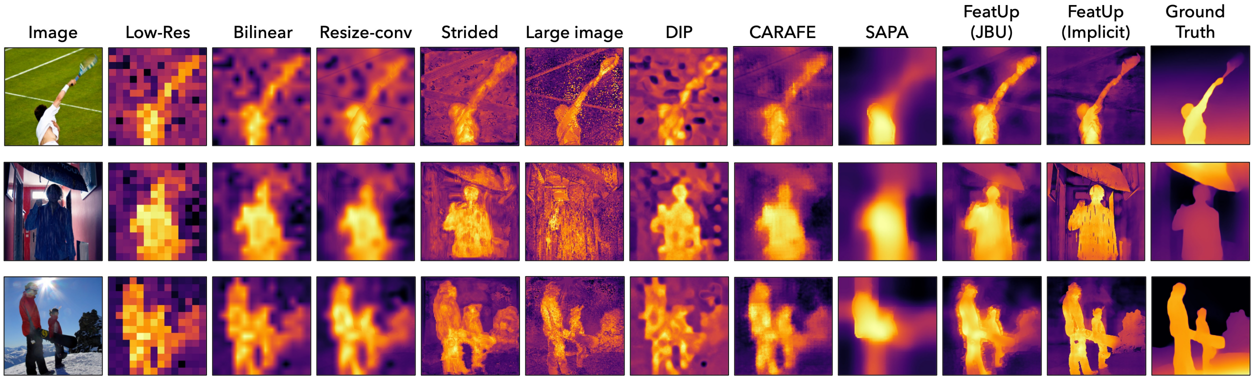


Figure C.13: Examples of linear probe transfer learning for depth estimation. Our methods produce sharper object boundaries and smoother interiors that more closely align with true depth than other methods.

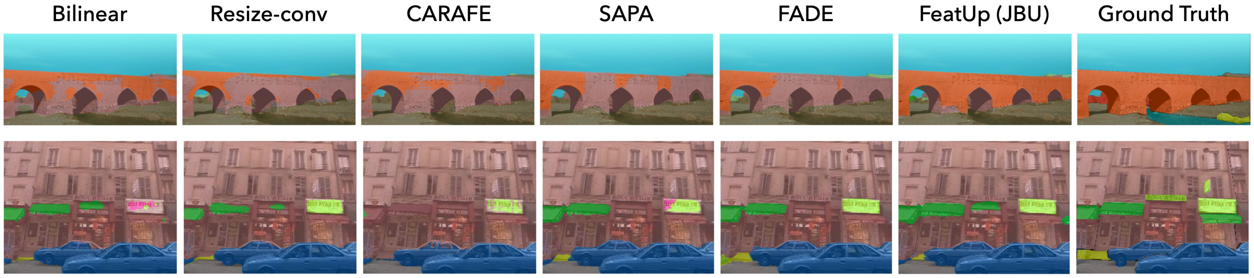


Figure C.14: End-to-end training performance of different upsampling methods from our Segformer based semantic segmentation experiments. These results do not use linear probes, but instead train the architecture jointly.

C.14 Limitations

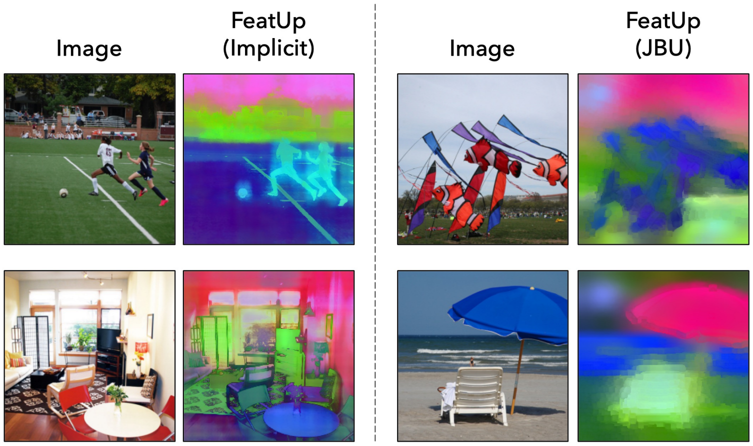


Figure C.15: Left: Though FeatUp's implicit network can capture fine detail such as the soccer ball or window frame, it can still produce some halo effects (see soccer player). Additionally, because the method relies on the input image's spatial signal, certain patterns unrelated to object semantics can be transferred to the feature map (see rug pattern), though this is a rare occurrence. Right: FeatUp's JBU network is not as sensitive to fine detail as the implicit network, instead capturing broader contours.

C.15 Implementation Details

All backbones (DINO, DINOv2, ViT, ResNet-50, CLIP, and DeepLabV3) used to train FeatUp are frozen, pre-trained models obtained from the community. We outline the hyperparameters used to train FeatUp in table C.5.

Hyperparameter	**FeatUp (Implicit)**	**FeatUp (JBU)**
Num Images	1	4
Num Jitters Per Image	10	2
Downsampler	Attention	Attention
Optimizer	NAdam	NAdam
Learning Rate	0.001	0.001
Image Load Size	224	224
Projection Dim	128	30
Training Steps	2000	2000
Max Transform Padding	30px	30px
Max Transform Zoom	1.8×	2×
Kernel Size	29	16
Total Variation Weight	0.05	0.0
Implicit Net Layers	3	n/a
Implicit Net Dropout	0.1	n/a
Implicit Net Activation	ReLU	n/a

Table C.5: Hyperparameters used in training FeatUp.

.

Appendix D

Appendix for Chapter 6

D.1 Full Cross Modal Retrieval Results

	Places Audio Retrieval						AudioSet Retrieval					
	I → A			A → I			I → A			A → I		
Method	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
[274]	12.1%	33.5%	46.3%	14.8%	40.3%	54.8%	-	-	-	-	-	-
[254]	13.0%	37.8%	54.2%	16.1%	40.4%	56.4%	-	-	-	-	-	-
DAVENet [231]	12.7%	37.5%	52.8%	20.0%	46.9%	60.4%	-	-	-	-	-	-
DAVENet* [231]	13.3%	38.3%	51.2%	20.5%	45.3%	57.2%	0.10%	0.70%	1.30%	0.10%	0.30%	1.20%
CAVMAE*[229]	36.7%	70.3%	81.7%	33.9%	65.7%	77.7%	22.8%	44.9%	55.7%	21.1%	41.7%	50.7%
ImageBind[247]	0.10%	0.50%	1.10%	0.10%	0.40%	1.10%	29.6%	55.4%	64.5%	31.8%	57.3%	66.5%
Ours	**65.3%**	**90.0%**	**94.2%**	**64.4%**	**89.4%**	**94.3%**	**35.1%**	**58.0%**	**68.2%**	**33.6%**	**59.3%**	**68.4%**

Table D.1: Full cross modal retrieval results using the same setting of Table 6.2. We note DenseAV outperforms all baselines in all metrics and all datasets.

D.2 VGGSound Source Evaluation

Table D.2 adds evaluations on the VGGSound Source dataset. We note that VGGSS annotation's large bounding boxes do not reward high-resolution results. Nevertheless, DenseAV outperforms all methods including 5 additional baselines (Attention10K [390], AVObject [391], LVS [242], FNAC AVL [392], and SLAVC [393]).

Method	cIoU	AUC
DAVENet	6.8%	21.2%
CAVMAE	7.9%	25.0%
ImageBind	3.4%	20.5%
Attention10K	18.5%	30.2%
AVObject	29.7%	35.7%
LVS	34.4%	38.2%
SLAVC	38.8%	38.8%
FNAC AVL	39.4%	39.4%
Ours	**40.6%**	**40.6%**

Table D.2: Performance on VGGSound Source localization.

D.3 Speech Prompted Semantic Segmentation Noise Robustness:

DenseAV was trained with natural speech and sounds and is robust to environmental noise and common speech errors like stutters. We explore additional noise-robustness experiments in Table D.3.

Method	mAP	mIoU
DAVENet	31.8%	26.1%
CAVMAE	27.2%	23.8%
ImageBind	20.2%	19.7%
Ours	**48.1%**	**36.6%**

Table D.3: Performance on speech based semantic segmentation task with environmental noise from the MUSAN [394] dataset added to spoken category labels.

D.4 Speech Prompted Semantic Segmentation Examples

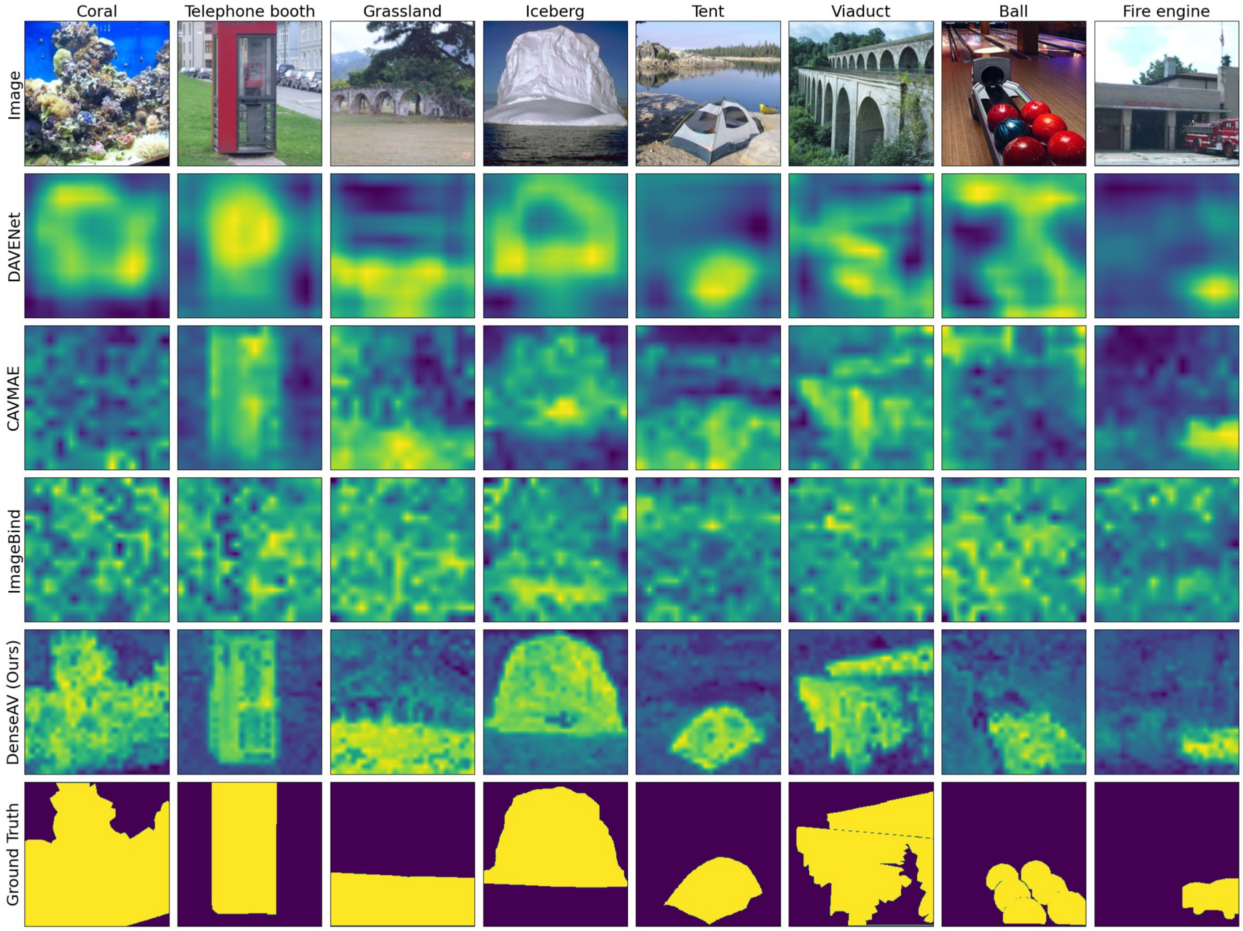


Figure D.1: Selected visualizations of AV heatmaps for the *speech prompted* semantic segmentation task. We visualize results across several baselines. DenseAV achieves the best localization performance both qualitatively and quantitatively, highlighting the full extent of objects with high resolution heatmaps.

D.5 Sound Prompted Semantic Segmentation Examples

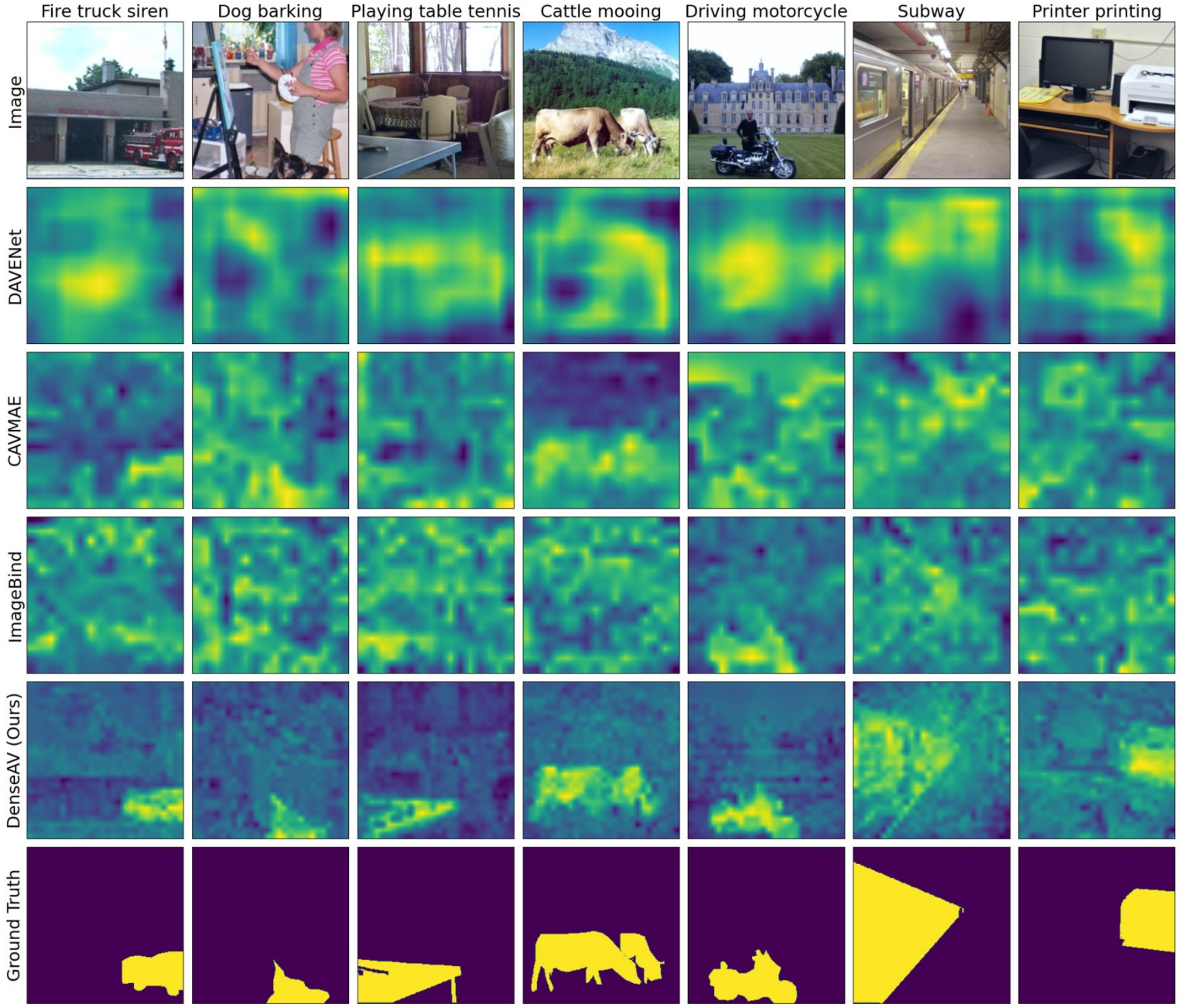


Figure D.2: Selected visualizations of AV heatmaps for the *sound prompted* semantic segmentation task. We visualize results across several baselines. DenseAV achieves the best localization performance both qualitatively and quantitatively, highlighting the full extent of objects with high resolution heatmaps. We note that DenseAV can highlight objects even if they are not centered or clearly visible as in the dog example (second column).

D.6 Comparison Across Backbones

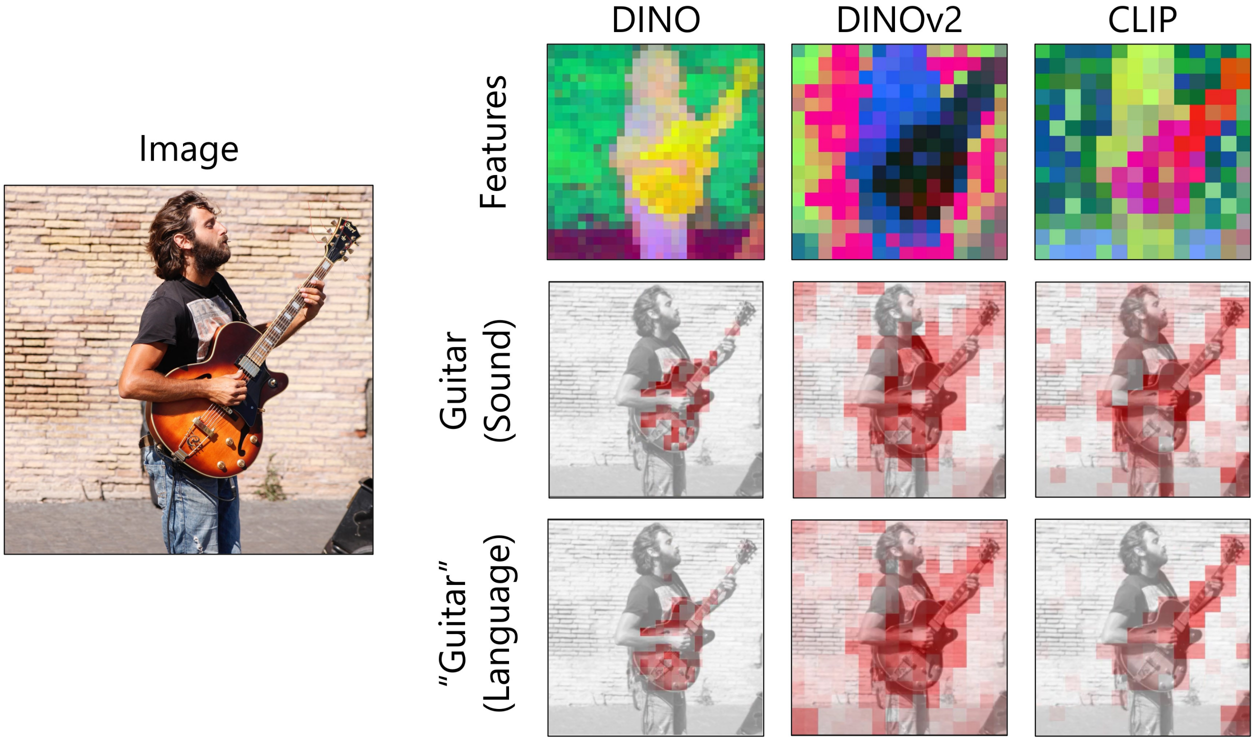


Figure D.3: Comparison of sound and speech prompted localization of DenseAV with various choices of visual backbone. DINO's features are both the best for localization as well as the highest resolution because of its 8 × 8 patch size.

D.7 Associating Spoken Words to Visual Objects

Visual Object	Top 5 Retrieved Words				
ottoman	sofa	chair	chair	seat	living
ruins	brick	stone	castle	clay	stone
dirt track	dirt	dirt	trail	field	dirt
monitor	screen	screen	computer	television	screen
control panel	cockpit	airplane	cockpit	airplane	airplane
bar	desk	picture	counter	poker	kitchen
waterfall	waterfall	fountain	water	waterfall	waterfall
embankment	trench	land	field	land	hill
bleachers	amphitheater	steps	colosseum	step	stairway
snow	snow	snow	snow	mountain	snow

Table D.4: Top 5 word retrieval using DenseAV's visual object features on the speech prompted semantic segmentation dataset described in Section 6.6.2. We determine if DenseAV can perform fine-grained speech retrieval by seeing if inner activations properly highlight the definitions of objects. We average visual features of visual objects to form a visual object query vector. We then form word representations for the PlacesAudio validation set by averaging speech features over an utterance using word timing information provided by Microsoft's Speech to Text API. Feature averaging strategy is depicted in Figure D.4. For each visual object, we retrieve the top 5 nouns from the PlacesAudio spoken captions. We do not average across words, so if a word appears twice in the table it represents two different spoken instances. Some visual objects are able to retrieve instances of speech that directly correspond to the name of the object, such as snow and waterfall. Others retrieve a variety of relevant words for example the "ruins" visual object retrieves instances of people saying "brick", "castle", and "stone". We note that the 10 visual objects selected were randomly selected from the hundreds in our speech prompted semantic segmentation dataset.

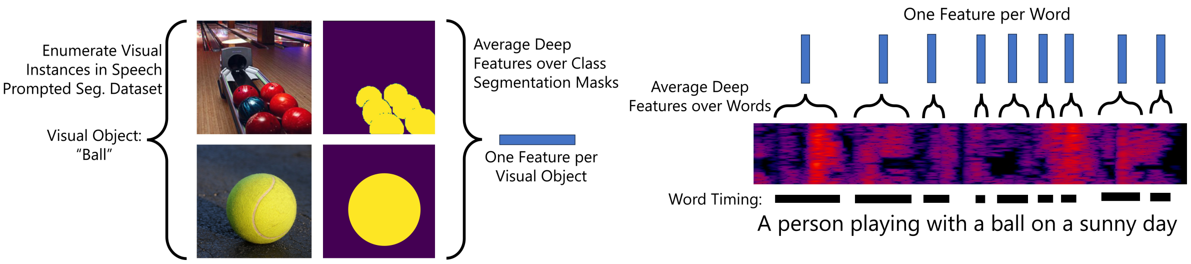

Enumerate Visual Instances in Speech Prompted Seg. Dataset
Visual Object: "Ball"
Average Deep Features over Class Segmentation Masks
One Feature per Visual Object
One Feature per Word
Average Deep Features over Words
Word Timing:
A person playing with a ball on a sunny day

Figure D.4: Diagram of feature averaging strategy used for the retrieval experiment in Table D.4. We average visual features over all instances of a visual object as shown in the left hand side, using the segmentation mask to only include visual features for the object of interest. To form features for each word in the places audio dataset, we use word timing information to average deep features over the extent of an utterance. Once we form features for all visual objects and all words, we retrieve the top 5 nouns for each visual object.

D.8 Failure Cases

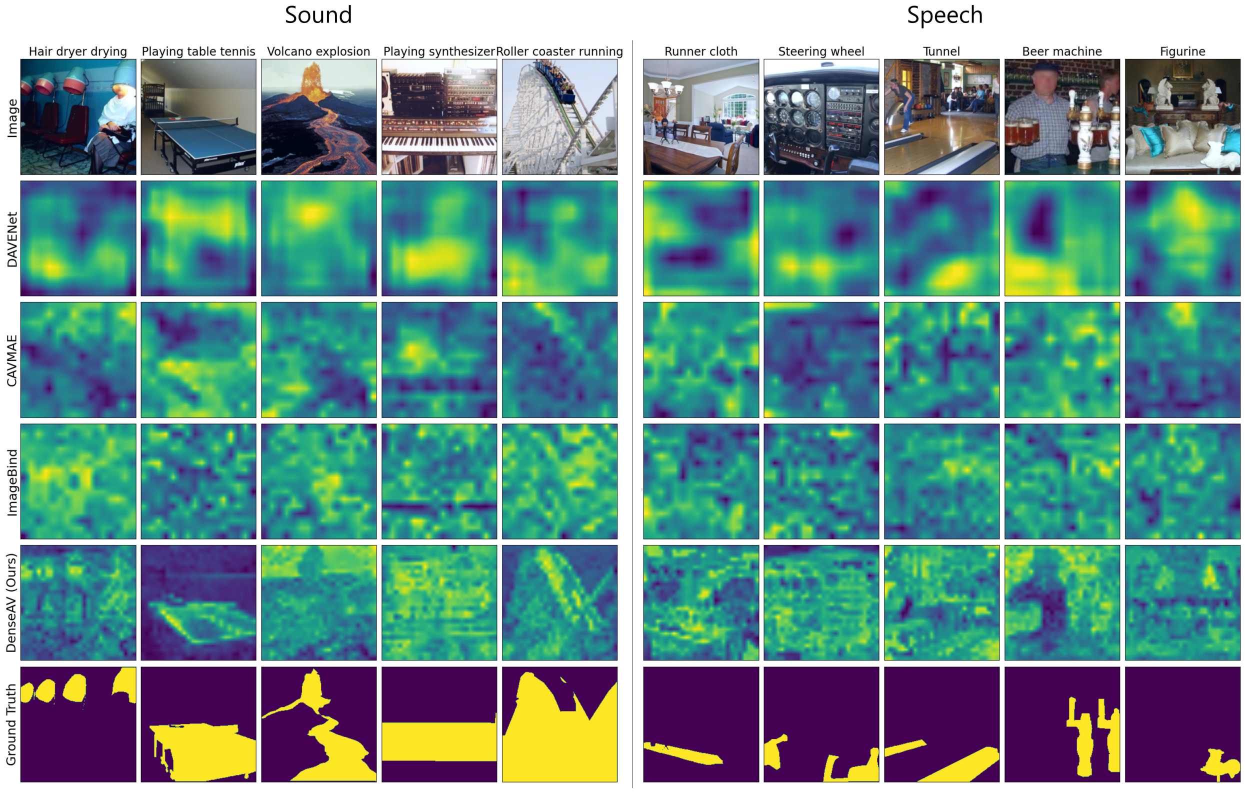


Figure D.5: Examples of DenseAV's failure cases on speech and sound prompted semantic segmentation. On unusual visual objects such as the "hair dryer drying" activations are more diffuse than other hair dryers in the dataset, likely because of its rarer form. A similar effect appears in the steering wheel example likely because steering wheel is often infrequently used to describe airplane controls. Rare sounds like volcano explosions, or rare visual obnjects like the bowling "tunnels" cause similar diffuse activations. Like many discriminitive algorithms, DenseAV has some tendency to bias towards discriminitive regions such as the top of the table tennis board in the "playing table tennis". There is also some mismatch between ADE20K labels and what you might expect a reasonable algorithm should highlight, as evidenced by the "roller coaster running" sound example. Similarly in the "Figurine" example, the algorithm reasonably associates figurines with the lions in the background instead of the dog in the foreground. Finally the beer machine example shows how there's some ambiguity between whether an algorithm should respond to compound words and ideas. Should it couple "beer" to the beer glass and "machine" to the spigots, or should "beer-machine" entirely couple to the spigots. DenseAV seems to choose the former as the beer in the foreground and background is also activated in this speech prompted example.)

D.9 Comparing to DINO CLS Token Activations

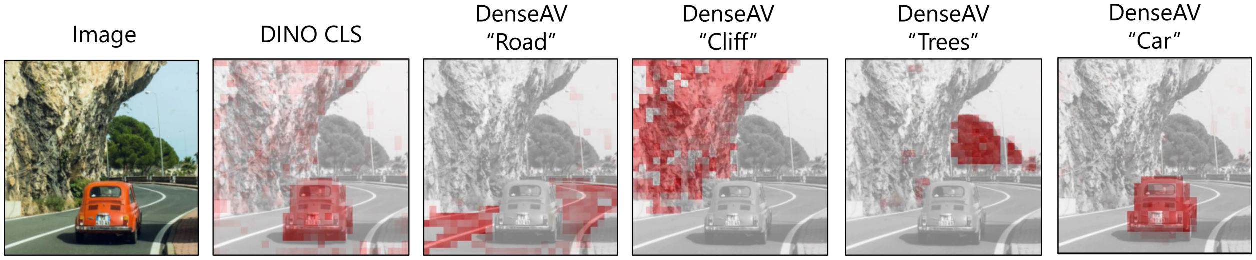


Figure D.6: Comparison of DINO CLS token heatmap visualization [226] and DenseAV's activations. DenseAV does not just select salient objects as DINO's CLS token does. Instead, within a single video clip DenseAV can highlight the meaning of words as they are spoken. Depending on the word spoken, this can accurately highlight a variety of objects in the scene, even if they are less salient like the trees in the background.

D.10 Visualizing Activations when an Object is not Present

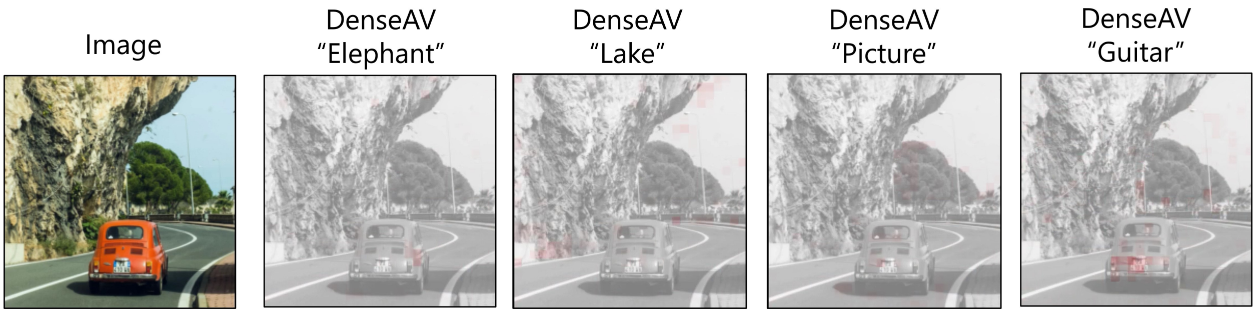


Figure D.7: Visualization of DenseAV activations when an object is not present in a scene. DenseAV's activations are significantly smaller than when objects are present in a scene like in Figure D.6.

D.11 Additional Regularizer Details

Negative Audio Splicing Though using 6.3 is enough to make a reasonable cross-modal retrieval system, the extreme flexibility of self-attention operator in modern transformers can lead to degenerate solutions. For example, we found that without regularizers that encourage local features to be meaningful, the network could develop its own "global" tokens by selecting a handful of local tokens to carry all of the information. This is similar to the observation of [268] and we observed this occasionally in our audio branch, which would collapse to only use the first tokens. To keep the network from collapsing the semantics of the audio clip into a single token, we introduce small negative sample clips into our audio samples. These small negative audio regions are randomly spliced into the larger audio clip, and we encourage the network to set the couplings in these regions to zero with a l_2 regularizer. We include further details of the DenseAV's architecture, hyperparameters, and regularizers in the Supplement.

More formally, let $(a_b, v_b)_1^B$ be a **B**atch of B paired audio and visual signals as before. Let $m_b \in [0, 1]^T$ be a soft mask where that measures whether a given location in the audio signal is actually part of a spliced negative clip. For example, $m_b[t] = 1$ when the clip at time t is part of the negative clip, $m_b[t] = 0$ in the positive part of the clip, and $0 < m_b[t] < 1$ in the small boundary regions when the true clip is being spliced into the negative clip and both sounds are present. Our negative audio splicing regularizer squares each entry of the similarity tensor and averages these according to the strength of the negative clip indicator m_b:

$$\mathcal{L}_{Splice} = \text{WeightedMean}(s(a_b, v_b)^2, m_b) \tag{D.1}$$

Where the mean assumes that the weighting strength m_b has been broadcast to the shape of $s(a_b, v_b)^2$. We point interested readers to the supplement for explicit formulations of these regularizers which are too verbose for the double-column format here. Intuitively, this term penalizes the network for having activations during a period of spliced negative audio. We also note that we apply this regularizer to any padded silence at the ends of short audio clips.

Calibration Regularization The calibration temperature provides the network with the crucial ability to increase or decrease its certainty by updating a single parameter. However, the network can also achieve this effect by increasing or decreasing the magnitudes of its features. We found that sometimes the temperature would accelerate downward, forcing the feature magnitudes to increase to compensate. As a result, the network would eventually saturate or become unstable. We hypothesize that this is due to optimizer momentum, and we prevent this "runaway calibration", by adding a small regularizer to the temperature parameter γ

$$\mathcal{L}_{Cal} = \max(\log(1) - \log(\gamma), 0)^2 \tag{D.2}$$

This term penalizes the calibrator when it drops below 1.0 and encourages the calibrator to stay at or above 1.0.

Nonnegative Pressure The InfoNCE loss function is invariant to the addition of a scalar to every inner product. Thus, to the network can choose to either find evidence of "positive" couplings connecting similar objects or "negative" couplings connecting regions that definitely do not belong together. We found that by encouraging the network to look for "positive" evidence, as opposed counterfactual evidence, improved training stability and performance across the key metrics we investigate. To encourage this behavior, we add a small regularizer to encourage inner products between features to be ≥ 0. More specifically, let Ω be a set of 250 randomly selected coordinates (b, b', k, f, t, h, w). We then form our non-negativity regularizer:

$$\mathcal{L}_{NonNeg} = \frac{1}{|\Omega|} \sum_{\Omega} \min\left(s(a_b, v_{b'})[k, f, t, h, w], 0\right)^2 \tag{D.3}$$

This regularizer penalizes the similarity tensor if it drops below zero, encouraging features to exhibit positive couplings. We note than other works [253], have noted the benefits of using only non-negative feature couplings.

Disentangement Regularization DenseAV's multi-head similarity aggregation allows the network to use its different heads to model different independent ways that the audio and video modalities could couple together. Interestingly we find that if we give DenseAV two heads, one naturally specializes to language and the other head to more generic sounds. In particular, we find that one head will rediscover the meaning of words by "grounding" them to visual objects and another head will localize which objects created a given sound. To purify this disentanglement of concepts without supervision, we encourage different attention heads of our algorithm to specialize. More specifically we penalize the network when multiple attention heads are simultaneously active. In our experiments we use two attention heads. As before, we let $(a_b, v_b)_1^B$ be a **B**atch of B paired audio and visual signals. Our disentanglement loss for two heads is then:

$$\mathcal{L}_{Dis} = \text{Mean}(|s(a_b, v_b)[1] \circ s(a_b, v_b)[2]|) \tag{D.4}$$

Where $\circ$ represents elementwise multiplication and $|\cdot|$ is the elementwise absolute value function. $[k]$ mirrors PyTorch slicing notation and refers to selecting the activations for only the kth attention head. Intuitively, this loss will encourage one head to be silent if the other head is active and can be viewed as a "cross-term" generalization of the l^2 regularizer [272] for encouraging activation shrinkage.

Total Variation Smoothness To improve the quality and temporal consistency of discovered audio-visual couplings we impose a smoothness regularizer, $\mathcal{L}_{TV}$, in the audio-time dimension.

$$\mathcal{L}_{TV} = \text{Mean}((\text{act}(1 : t-1) - \text{act}(2 : t))^2) \tag{D.5}$$

Where the activations for a given time slice $[1, t-1]$ are given by:

$$\text{act}(1 : t-1) = (s(a_b, v_b)[:, :, t', :, :])_{t'=1}^{t-1} \tag{D.6}$$

Informally, this regularizer penalizes when the inner product strengths change quickly over time.

Full Stability Regularizer Putting these terms together into a single equation we have:

$$\mathcal{L}_{Stability} = \lambda_{Splice}\mathcal{L}_{Splice} + \lambda_{Cal}\mathcal{L}_{Cal} + \lambda_{NonNeg}\mathcal{L}_{NonNeg} + \lambda_{TV}\mathcal{L}_{TV} \tag{D.7}$$

Where $\lambda_{Splice} = 0.01$, $\lambda_{Cal} = 0.1$, $\lambda_{NonNeg} = 0.01$, and $\lambda_{TV} = 0.01$.

D.12 Regularizer Ablation

Regularizer				Speech Semseg.		Places Acc. @ 10	
$\mathcal{L}_{Cal}$	$\mathcal{L}_{NonNeg}$	$\mathcal{L}_{Splice}$	$\mathcal{L}_{TV}$	mAP	mIoU	$I \rightarrow A$	$A \rightarrow I$
✓	✓	✓	✓	48.7%	36.8%	94.2%	94.3%
	✓	✓	✓	49.1%	37.3%	94.3%	94.1%
✓		✓	✓	48.2%	36.8%	94.1%	93.4%
✓	✓		✓	48.6%	36.7%	94.8%	94.5%
✓	✓	✓		49.0%	36.9%	94.2%	93.7%
				-	-	-	-

Table D.5: Ablation study of the different components of $\mathcal{L}_{Stability}$. We find that on the whole $\mathcal{L}_{Stability}$ is needed to avoid collapse as shown the the bottom row of the table. However, removing any individual term does not have much effect on the final metrics.

Appendix E

Appendix for Chapter 7

E.1 Video and Code

We include a short video description of our work at https://aka.ms/axiomatic-video.

We also provide training and evaluation code at https://aka.ms/axiomatic-code

E.2 Evaluation and Implementation Details

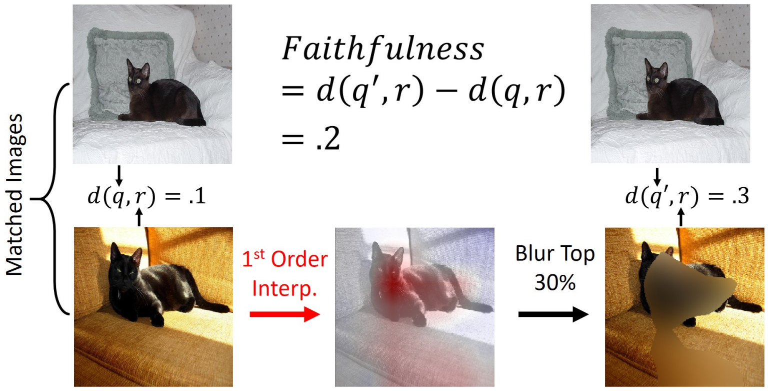


Figure E.1: First-order interpretation evaluation strategy. A good method should highlight pixels in the query image (top left and middle) that, when censored (top right), have the largest possible impact on the cosine distance.

Models: Our evaluation experiments use visual similarity systems built from "backbone" networks that featurize images and compare their similarity using cosine distance. We consider supervised backbones and contrastive unsupervised backbones. In particular, ResNet50 [121], VGG11 [334], and DenseNet121 [333] are trained with human classification annotations from the ImageNet dataset [134] and MoCo V2 is trained using unsupervised contrastive learning on ImageNet. We use torchvision [395] based model implementations and pre-trained weights except for MocoV2 which we download from [396] (800 epoch model). For kernel convergence experiments in Figure 7.5 we use randomly initialized three layer deep networks with Glorot [211] initialization, rectified linear unit activations, and a 20 dimensional hidden layer. We note that the functional form is not of much importance for these experiments so long as the function is nonlinear and non-quadratic. We provide an additional example in Figure E.4 on random 15 dimensional Boolean functions formed by enumerating and summing all possible variable products and weighting each by a uniform coefficient between 0 and 10.

Data: For evaluations within Table 7.1 we use the Pascal VOC [331] dataset. In particular we form a paired image dataset by using MoCo V2 to featurize the training and validation sets. All experiments use images that have been bi-linearly resized to 224 × 224 pixels. For each image in the PascalVOC validation set we choose a random segmentation class that contains over 5% of image pixels. We then find each validation image's closest "Conditional Nearest Neighbor" [332] from the images of the training set of the chosen segmentation class. We use cosine similarity between MoCoV2 deep features to find nearest neighbors. With this dataset of pairs, we can then compute our first and second order evaluation metrics. We provide instructions for downloading the pairs of images in the attached code. We note that our approach for selecting pairs of images with matching segmentation labels allow for measuring Faithfulness and success in label propagation as measured by mIoU.

Metrics: Our attached code contains implementations all metrics for preciseness but we include descriptions of metrics here for clarity. To measure first order faithfulness, we take a given validation image and training image from our dataset of paired images and compute the first order heat-map over the validation image. We then blur the top 30% of pixels by blurring the image with a 25 × 25 pixel blur kernel and replacing the top 30% of original image pixels with those from the blurred image. The drop in cosine similarity between the unblurred images and the unblurred training and blurred validation image is the first order faithfulness. We illustrate our first-order evaluation strategy in Figure E.1.

For our second-order evaluation, we use the ground truth semantic segmentation mask of the training image as a "query" attention signal. We then use the second-order interpretation methods to "project" this attention to the "retrieved" validation image. We censor all but the most-attended pixels in the retrieved image. The size of the remaining pixels matches the size of the validation image's selected semantic segmentation mask. In the second-order case we additionally measure the mean intersection over union (mIoU) of the resulting mask compared to the ground-truth retrieved image segmentation. A good approach should attend to jut the pixels of the segmentation class and thus yield a mIoU of 1 (maximum value) as a binary segmentation problem. We illustrate our second-order evaluation strategy in Figure 7.6.

Finally, for those methods that permit it, we measure how much they violate the efficiency axiom by summing the interpretation coefficients and comparing with $v(N) - v(\emptyset)$. In the first order setting $v(N)$ is the similarity between query and retrieved image, and $v(\emptyset)$ is the similarity between query and a blurred retrieved image (with 25 pixel blur). In the second order setting $v(\emptyset)$ represents the similarity when both images are blurred. For SAM-based methods we replace features with those from blurred images. To compute the sum of interpretation coefficients for kernel methods we sum over Shapley values in the first order case and over Shapley-Taylor indices of order $k \leq 2$ in the second-order case. For Partition SHAP [325] we sum coefficients over all pixels. For Integrated Hessian's we sum over all first and second order coefficients as described in [324].

In tables we report mean values of Inefficiency, and Faithfulness metrics and note that for these experiemtns the Standard Error of the Mean (SEM) is far below the three significant figure precision of the table.

First Order Methods: For first order explanations we use the official implementation of ImageLIME [397] and use the SHAP package for Integrated Gradients, Partition SHAP, and Kernel SHAP [325]. We re-implement SBSM and VESM in PyTorch from the instructions provided in their papers. For sampling procedures such as LIME, Kernel SHAP, and Partition SHAP we use 5000 function evaluations. For first and second-order super-pixel based methods (LIME, Kernel-SHAP) we use the SLIC superpixel method [398] provided in the Scipy library [399] with 50 segments, $compactness = 10$, and $\sigma = 3$. For SBSM we use a window size of 20 pixels and a stride of 3 pixels. We batch function evaluations with minibatch size 64 for backbone networks and 64×20 for SAM based methods. For all background distributions we blur the images with a 25-pixel blur kernel with the exception of LIME and SBSM which use mean color backgrounds.

Second Order Methods: For second order methods we use the same background and superpixel algorithms, but implement all methods within PyTorch for uniform comparison. For SBSM, Kernel SHAP, and LIME we use 20000 samples and for KSAM and IGSAM we use 40000 samples. For IGSAM we use the expected Hessians method referenced in the supplement of [324]. We use the PyTorch "lstsq"function for solving linear systems. For more details on our generalization of SBSM see Section E.11.

Compute and Environment: Experiments use PyTorch [131] v1.7 pre-trained models, on an Ubuntu 16.04 Azure NV24 Virtual Machine with Python 3.6. For all methods that require many network evaluations we use PyTorch DataLoaders with 18 background processes to eliminate IO bottlenecks. We standardize experiments using Azure Machine Learning and run each experiment on a separate virtual machine to avoid slowdowns due to scarce CPU or GPU resources.

E.3 Proof of Proposition 7.6.2

Let $v(S) : [0,1]^N \to \mathbb{R} := f(mask(x, S))$ represent soft masking of the spatial locations of a deep feature map x with the vector of zeros and applying a differentiable function f. We begin with the formulation of Integrated Gradients:

$$\mathrm{IG}_{hw}(S) = (S_{hw} - S'_{hw}) \int_{\alpha=0}^{1} \frac{\partial v(\alpha S + (1-\alpha)S')}{\partial T_{hw}}$$

In our case the foreground, $S := \mathbb{1}^{HW}$, is a mask of all 1s and the background, S', is the zero mask of the same shape. We note that the $\frac{\partial}{\partial T_{hw}}$ refers to taking the partial of the full input αS, not just the mask S. We include this to stress the subtle difference which can be missed in a quick reading of the equations of [335]. In this case our formula is simplified to:

$$\mathrm{IG}_{hw}(S) = \int_{\alpha=0}^{1} \frac{\partial v(\alpha S)}{\partial T_{hw}}$$

Approximating this integral with a single sample at $\alpha = 1$ yields:

$$\begin{aligned}
\mathrm{IG}_{hw}(S) &\approx \frac{\partial v(S)}{\partial S_{hw}} \\
&= \frac{\partial f(mask(x, S))}{\partial T_{hw}} \\
&= \frac{\partial}{\partial T_{hw}} f(x \odot S) \\
&= \sum_c x_{chw} \frac{\partial f(x)}{\partial x_{chw}} \\
&= \sum_c x_{chw} GAP(\nabla_x f(x)) && \text{(Spatially Invariant Derivatives)}
\end{aligned}$$

Which is precisely the formulation of GradCAM. This also makes it clear that the global average pooling of GradCAM causes the method to deviate from integrated gradients in the general case. To construct a function where GradCAM violates the dummy axiom we simply have to violate the spatial invariance of gradients. We provide a specific example of this violation in E.4.

E.4 GradCAM Violates the Dummy Axiom

It is straightforward to construct examples where GradCAM violates the dummy axiom. For example, consider the function:

$$d(x, y) = sim_{cosine}(GAP(x), GAP(y \odot M))$$

Where sim_{cosine} represents cosine similarity, $\odot$ represents elementwise multiplcation, and $M \in [0, 1]^{CHW}$ is a mask where $M_{chw} = 0$ if $w \leq \frac{W}{2}$ and $M_{chw} = 1$ otherwise. Intuitively, M removes the influence of any feature on the left of the image making these features "dummy" features for the model. Because GradCAM spatially averages the gradients prior to taking the inner product with the feature map all features are treated equally regardless of how they are used. In this example, depicted in Figure E.2, positive contributions from the right side of the image are extended to the left side of the image despite the fact that the mask, M stops these features from impacting the prediction. Using a Shapley or Aumann-Shapley approximator on the feature space does not suffer from this effect as shown in the two right columns of Figure E.2.

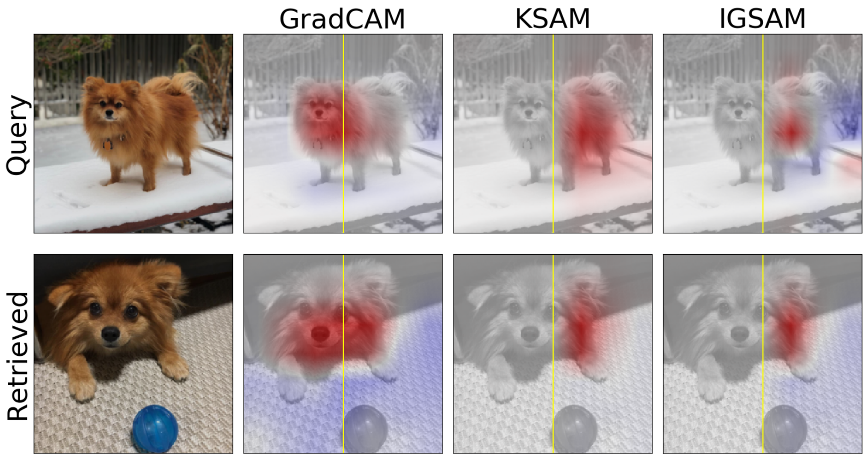


Figure E.2: Interpretations of a function that purposely ignores the left half of the image. KSAM and IGSAM properly assign zero weight to these features. GradCAM does not and hence violates the dummy axiom of fair credit assignment.

E.5 Integrated Gradient CAM

Sections E.4 and E.3 demonstrate that GradCAM can violate the dummy axiom when the function has spatially varying gradients which is a common occurrence especially if one is trying to interpret deeper layers of a network. We remedy this by instead considering Integrated Gradients on a function which masks the spatial locations of a deep feature map. More specifically our Integrated Gradient generalization of CAM takes the following form:

$$IGCAM(h,w) := \int_{\alpha=0}^{1} \frac{\partial f(b + \alpha M \odot (x - b))}{\partial T_{hw}} \tag{E.1}$$

Where f is the classification "head", $x \in \mathbb{R}^{CHW}$ is a tensor of deep image features, $M := \mathbb{1}^{HW}$ is a mask of 1s over the spatial location of the features, $b \in \mathbb{R}^{CHW}$ is a background signal commonly taken to be zero in GradCAM. We note that the $\frac{\partial}{\partial T_{hw}}$ refers to taking the partial of the full input $b + \alpha M \odot (x - b)$, not just the mask. We include this to stress the subtle difference which can be missed in a quick reading of the equations of [335]. This variant of GradCAM does not violate the dummy axiom and satisfies the axioms of the Aumann-Shapley fair credit assignment.

E.6 Additional Similarity Visualizations

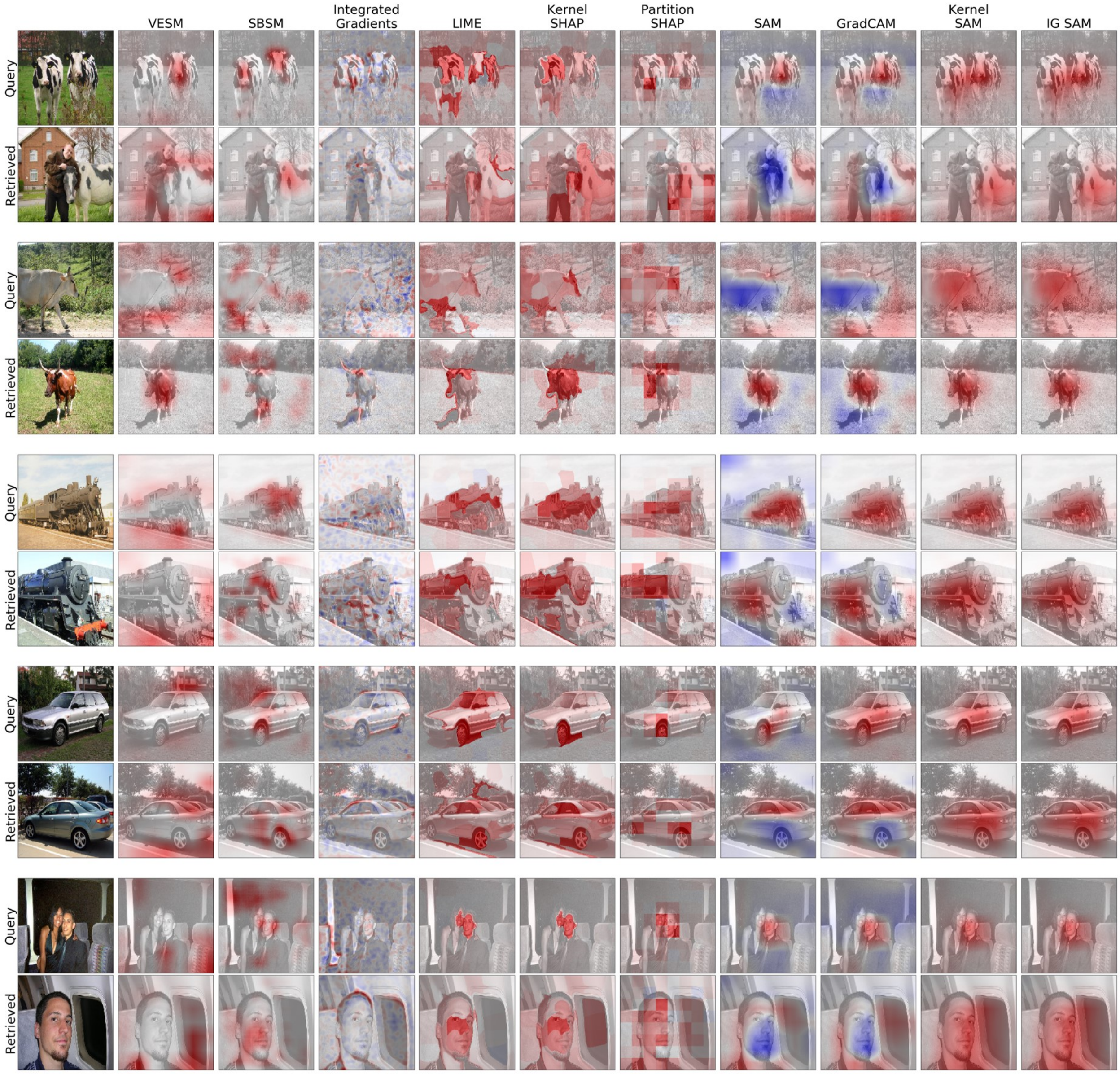


Figure E.3: Additional first-order search interpretations on random image pairs from the Pascal VOC dataset

E.7 Additional Results for Stanford Online Products

Table E.1: Comparison of performance of first-order search interpretation methods across different visual search systems on the Stanford Online Product dataset. Methods introduced in this work are highlighted in pink. *Though SAM generalizes [287] we refer to it as a baseline. For additional details see Section 7.8

		SBSM	**PSHAP**	**LIME**	**KSHAP**	VESM	GCAM	SAM*	**IG SAM**	**KSAM**
Metric	Model	Model Agnostic				Architecture Dependent				
Faith.	DN121	0.18	**0.23**	0.20	0.22	0.09	0.13	0.12	**0.18**	**0.18**
	MoCoV2	0.24	**0.30**	0.27	0.18	0.14	0.2	0.21	**0.24**	**0.24**
	RN50	0.11	**0.14**	0.12	0.13	0.03	0.07	0.07	**0.10**	**0.10**
	VGG11	0.15	**0.16**	0.14	0.15	0.04	0.08	0.09	**0.12**	**0.12**
Ineff.	DN121	-	**0.00**	0.24	**0.00**	-	11.2	0.54	0.02	**0.00**
	MoCoV2	-	**0.00**	0.17	**0.00**	-	0.34	0.57	0.02	**0.00**
	RN50	-	**0.00**	0.21	**0.00**	-	13.6	0.39	0.02	**0.00**
	VGG11	-	**0.00**	0.24	**0.00**	-	4.13	0.47	0.04	**0.00**

E.8 Additional Results for Caltech-UCSD Birds 200 (CUB) Dataset

Table E.2: Comparison of performance of first-order search interpretation methods across different visual search systems on the CUB dataset. Methods introduced in this work are highlighted in pink. *Though SAM generalizes [287] we refer to it as a baseline. RN50-ML refers to a ResNet50 architecture trained for metric learning on the CUB dataset with the margin loss [400]. For additional details see Section 7.8

		SBSM	**PSHAP**	**LIME**	**KSHAP**	VESM	GCAM	SAM*	**IG SAM**	**KSAM**
Metric	Model	Model Agnostic				Architecture Dependant				
Faith.	DN121	0.25	**0.38**	0.31	0.34	0.15	0.10	0.12	**0.30**	**0.30**
	RN50-ML	0.39	**0.49**	0.47	**0.49**	0.04	0.14	0.17	**0.41**	**0.41**
	MoCoV2	0.32	**0.47**	0.39	0.41	0.26	0.26	0.26	**0.34**	**0.34**
	RN50	0.14	**0.21**	0.18	0.18	0.05	0.07	0.07	**0.14**	**0.14**
	VGG11	0.23	**0.31**	0.26	0.27	0.11	0.15	0.16	**0.23**	0.22
Ineff.	DN121	-	**0.00**	0.17	**0.00**	-	16.0	0.58	0.02	**0.00**
	RN50-ML	-	**0.00**	0.13	**0.00**	-	5.23	0.48	0.03	**0.00**
	MoCoV2	-	**0.00**	0.19	**0.00**	-	0.44	0.60	0.03	**0.00**
	RN50	-	**0.00**	0.15	**0.00**	-	15.5	0.43	0.02	**0.00**
	VGG11	-	**0.00**	0.17	**0.00**	-	4.25	0.54	0.05	**0.00**

E.9 Additional Results for MS COCO

Table E.3: Comparison of performance of first and second-order search interpretation methods across different visual search systems on the MSCOCO dataset. Methods introduced in this work are highlighted in pink. *Though SAM generalizes [287] we refer to it as a baseline. For additional details see Section 7.8

Metric	Order	Model	SBSM	**PSHAP**	**LIME**	**KSHAP**	VESM	GCAM	SAM*	**IG SAM**	**KSAM**
			Model Agnostic				Architecture Dependent				
Faithfulness	First	DN121	0.18	**0.24**	0.22	0.22	0.10	0.12	0.11	0.14	**0.17**
		MoCoV2	0.25	**0.37**	0.33	0.35	0.15	0.23	0.24	0.24	**0.28**
		RN50	0.10	**0.14**	0.12	0.12	0.04	0.07	0.07	0.07	**0.09**
		VGG11	0.14	**0.15**	0.14	0.14	0.05	0.09	0.1	0.10	**0.12**
	Second	DN121	0.49	-	**0.57**	**0.57**	-	-	0.5	**0.51**	0.49
		MoCoV2	0.73	-	**0.79**	**0.79**	-	-	0.77	0.77	**0.78**
		RN50	0.73	-	**0.78**	**0.78**	-	-	**0.75**	**0.75**	0.73
		VGG11	0.67	-	**0.73**	**0.73**	-	-	0.71	0.71	**0.72**
Inefficiency	First	DN121	-	**0.00**	0.22	**0.00**	-	12.3	0.6	0.02	**0.00**
		MoCoV2	-	**0.00**	0.11	**0.00**	-	0.46	0.66	0.02	**0.00**
		RN50	-	**0.00**	0.22	**0.00**	-	15.8	0.47	0.02	**0.00**
		VGG11	-	**0.00**	0.31	**0.00**	-	3.47	0.59	0.04	**0.00**
	Second	DN121	-	-	0.15	**0.01**	-	-	0.20	0.01	**0.00**
		MoCoV2	-	-	0.10	**0.01**	-	-	0.09	0.02	**0.00**
		RN50	-	-	0.07	**0.01**	-	-	0.07	0.02	**0.00**
		VGG11	-	-	0.11	**0.01**	-	-	0.19	0.04	**0.00**
mIoU	Second	DN121	0.50	-	**0.62**	0.61	-	-	0.62	**0.63**	0.52
		MoCoV2	0.52	-	**0.62**	0.61	-	-	0.64	**0.66**	0.60
		RN50	0.50	-	**0.62**	0.61	-	-	0.63	**0.65**	0.48
		VGG11	0.50	-	**0.62**	0.61	-	-	0.66	**0.67**	0.60

E.10 Additional Kernel Convergence Results

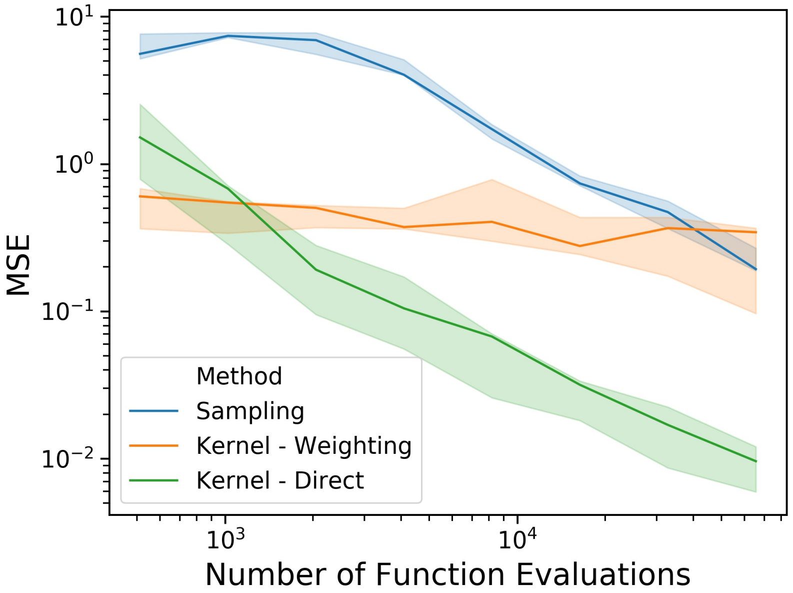


Figure E.4: Kernel convergence for random functions generated by randomly choosing coefficients. Results generally mirror those for randomly initialized deep networks

E.11 Generalizing SBSM to Second-Order Search Engine Interpretability

Before generalizing SBSM [289] to second-order interpretability we will review the original implementation for marginal interpretability. SBSM uses a sliding square mask and multiple evaluations of the search engine to determine which regions of the image are important for similarity. More formally, let q, and r represent the pixels of the query image and retrieved image. Let $M^s_{ij}(q)$ represent the result of replacing a square of pixels of size $s \times s$ centered at pixel (i, j) with a "background value" which in our case is black. SBSM "slides" this mask across the query image and compares the similarity between the masked query and retrieved image. These masked similarity values are compares to the baseline similarity value and stored in a weight matrix, w:

$$w_{ij} = \min \left[d\left(M^s_{ij}(q), r \right) - d\left(q, r\right), 0 \right] \tag{E.2}$$

Intuitively speaking, the weights w_{ij} represent the impact of masking a square centered at (i, j). For areas that are critical to the similarity, this will result in $w_{ij} > 0$. Finally, an attention mask on the query image is formed by a weighted average of the masks used to censor the images. For square masks, this can be achieved efficiently using a deconvolution with a kernel of ones of size $s \times s$ on the weight matrix w. We also note that instead of evaluating the (expensive) distance computation d for every pixel (i, j), one can also sample pixels to censor. We use this approach in our second-order generalization.

To generalize SBSM we use a pair of masks, one for the query image and one for the retrieved image respectively. We sample mask locations and calculate weights designed to capture the intuition that censoring corresponding areas cause similarity to increase as opposed to decrease. More specifically we use the following weighting scheme:

$$w_{ij}^{hw} = \min\left[d\left(q, r\right) - d\left(M_{ij}^{s}\left(q\right), M_{hq}^{s}\left(r\right)\right), 0\right] \tag{E.3}$$

Because evaluating the similarity function for every (i, j, h, w) combination is prohibitively expensive, we instead sample masked images for our computation. To project attention from a query pixel, we query for all masks that overlap with the selected query pixel, and then average their corresponding retrieved masks according to the weights calculated in Equation E.3.

E.12 Proof of Proposition 7.6.1

Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^{CHW}$ and represent the space of deep network features where C, H, W represent a channel, height, and width of the feature maps respectively. Let the function $d := \sum_c GAP(x)_c GAP(y)_c$. Let the grand coalition, $N = [0, H] \times [0, W]$, index into the spatial coordinates of the image feature map y. Let the function $mask(y, S)$ act on a feature map y by replacing the features at locations S with a background signal b. For notational convenience let $\psi_i(v) := \phi_v(i)$ represent the Shapley value the i^{th} player under the value function v. We begin by expressing the left-hand side of the proposition:

$$\begin{aligned}
\psi_{hw}(v) &= \psi_{hw}\left(\sum_c GAP(x)_c GAP(mask(y, S))_c\right) \\
&= \psi_{hw}\left(\frac{1}{HW}\sum_c GAP(x)_c \sum_{h'w'} mask(y, S)_{ch'w'}\right) \\
&= \frac{1}{HW}\sum_{h'w'} \psi_{hw}\left(\sum_c GAP(x)_c mask(y, S)_{ch'w'}\right) && \text{(Linearity)} \\
&= \frac{1}{HW}\psi_{hw}\left(\sum_c GAP(x)_c mask(y, S)_{chw}\right) && \text{(Dummy)} \\
&= \frac{1}{HW}\sum_c GAP(x)_c (y_{chw} - b_{chw}) && \text{(Efficiency)}
\end{aligned}$$

E.13 Proof of Proposition 7.7.1

Let the spaces $\mathcal{X}$, $\mathcal{Y}$ and function d be as in Proposition 7.6.1. As a reminder the function d represents the un-normalized GAP similarity function. Let the grand coalition, N, index into the spatial coordinates of both the query image features $x \in R^{CHW}$ and retrieved image features $y \in R^{CHW}$. Let the function $mask(y, S)$ act on a feature map y by replacing the corresponding features with a background feature map a for query features and b for retrieved features. We can represent the set of players, N, as a set of ordered pairs of coordinates with additional information about which tensor, the query (0) or retrieved (1) features, they represent:

$$N = ([1, H] \times [1, W] \times \{0\}) \cup ([1, H] \times [1, W] \times \{1\}) \tag{E.4}$$

In the subsequent proof we omit these 0, 1 tags as it is clear from our notation which side, query or retrieved, the index refers to based on the index h, w for the query and i, j for the retrieved image. We first consider the zero background value function, $v(S \subset N)$, defined by censoring the spatially varying features prior to global average pooling and comparing their inner product:

$$v(S) = \left(\frac{1}{HW}\sum_{h,w} \tilde{x}_{chw}\right) \cdot \left(\frac{1}{HW}\sum_{i,j} \tilde{y}_{cij}\right)$$

where

$$\tilde{x}_{chw} = \begin{cases} x_{chw} & (h, w) \in S \\ 0 & o.w. \end{cases}$$

and likewise, for y_{cij}. When S contains all i, j, h, w this represents the similarity judgement from the GAP network architecture. We seek the Shapley-Taylor index for a pair of image locations $S = \{(h, w), (i, j)\}$. For notational convenience let $\psi_S^k(v) := \phi_v^k(S)$ represent the $k-$order interaction effects for the subset S and the value function v.

$$\begin{aligned}
\psi_S^k(v) &= \psi_S^k\left(\left(\frac{1}{HW}\sum_{h',w'} \tilde{x}_{ch'w'}\right) \cdot \left(\frac{1}{HW}\sum_{i',j'} \tilde{y}_{ci'j'}\right)\right) \\
&= \psi_S^k\left(\frac{1}{H^2W^2}\sum_{c}\sum_{h',w'}\sum_{i',j'} \tilde{x}_{ch'w'}\tilde{y}_{ci'j'}\right) \\
&= \psi_S^k\left(\frac{1}{H^2W^2}\sum_{h',w'}\sum_{i',j'}\sum_{c} \tilde{x}_{ch'w'}\tilde{y}_{ci'j'}\right) \\
&= \sum_{h',w'}\sum_{i',j'} \psi_S^k\left(\sum_{c} \frac{1}{H^2W^2}\tilde{x}_{ch'w'}\tilde{y}_{ci'j'}\right) && \text{(Linearity)} \\
&= \psi_S^k\left(\sum_{c} \frac{1}{H^2W^2}\tilde{x}_{chw}\tilde{y}_{cij}\right) && \text{(Dummy)} \\
&= \psi_{S'}^k\left(v_{hwij}\right) && \text{(Renaming)}
\end{aligned}$$

Where the renaming of the last step was because we can now consider a simplified value function with just the non-dummy players as $v_{hwij}(S') := \sum_c \tilde{x}_{chw}\tilde{y}_{cij}$. Where S' represents a subset of the non-dummy players: $N' = \{(h, w), (i, j)\}$. We can now explicitly calculate the index:

$$\begin{aligned}\psi^2_{S'}(v) &= \frac{2}{n}\sum_{T\subseteq N'\setminus S'} \delta_{S'} v_{hwij}(T)\frac{1}{\binom{n-1}{t}} \\ &= \delta_{S'} v_{hwij}(\emptyset) \\ &= \frac{1}{H^2W^2}\sum_c x_{chw}y_{cij}\end{aligned}$$

By following the same set of reasoning, we can introduce nonzero background values a_{chw} and b_{cij} to yield the following:

$$\psi^2_{hw,ij}(v) = \frac{1}{H^2W^2}\sum_c x_{chw}y_{cij} - x_{chw}b_{cij} - a_{chw}y_{cij} + a_{chw}b_{cij} \tag{E.5}$$

E.14 Proof that Shapley Taylor is Proportional to Integrated Hessians for GAP architecture

As in Proposition 7.6.2 we consider the soft masking or "multilinear extension" of our second-order value function v_2:

$$v_2(S) : [0,1]^N \to \mathbb{R} := d(mask(x,S), mask(y,S)) \tag{E.6}$$

let hw, and ij be members of the grand coalition N such that $hw \neq ij$. We begin our proof with the expression for the off-diagonal terms of the Integrated Hessian.

$$\Gamma_{hw,ij}(S) := \int_{\alpha=0}^{1}\int_{\beta=0}^{1} \alpha\beta\frac{\partial^2 v_2(\alpha\beta S)}{\partial T_{hw}\partial T_{ij}} \tag{E.7}$$

Where $\frac{\partial}{\partial T_h w}$ represents the hw component of the partial derivative with respect to $\alpha\beta S$, not to be confused with the partial derivative of just S. Like in our proof of Proposition 7.6.2, because our function is defined on the interval $[0,1]^N$ many of the terms mentioned in [324] drop out and instead are captured in the Hessian of the function with repspect to the soft mask. We now expand the definition of $v_2(\alpha\beta S)$:

$$\begin{aligned}v_2(\alpha\beta S) &= d(mask(x,\alpha\beta S), mask(y,\alpha\beta S)) \\ &= \frac{1}{H^2W^2}\sum_c\left(\sum_{h,w} a_{chw} + \alpha\beta S_{hw}(x_{chw}-a_{chw})\right)\left(\sum_{i,j} b_{cij} + \alpha\beta S_{ij}(y_{cij}-b_{cij})\right)\end{aligned}$$

From this function we can read off the appropriate term of the hessian with respect to the mask at location (h,w) and location (i,j)

$$\begin{aligned}\frac{\partial^2 v_2(\alpha\beta S)}{\partial T_{hw}\partial T_{ij}} &= \frac{1}{H^2W^2}\sum_c x_{chw}y_{cij} - x_{chw}b_{cij} - a_{chw}y_{cij} + a_{chw}b_{cij} \\ &= \psi^2_{hw,ij}(v)\end{aligned}$$

We can now pull this outside the integral to yeild:

$$\begin{aligned}\Gamma_{hw,ij}(v_2) &= \int_{\alpha=0}^{1}\int_{\beta=0}^{1}\alpha\beta\frac{\partial^2 v_2(\alpha\beta S)}{\partial T_{hw}\partial T_{ij}}\\ &= \psi^2_{hw,ij}(v)\int_{\alpha=0}^{1}\int_{\beta=0}^{1}\alpha\beta\\ &= \frac{1}{4}\psi^2_{hw,ij}(v)\end{aligned}$$

Which proves that the Shapley-Taylor index and second order Aumann-Shapley values are proportional for the GAP architecture.

E.15 Explaining Dissimilarity

In addition to explaining the similarity between two images, our methods naturally explain image dissimilarity. In particular, regions with a negative Shapely values (Blue regions in Figure E.5) contribute negatively to the similarity between the two images. These coefficients can be helpful when trying to understand why an algorithm does not group two images together.

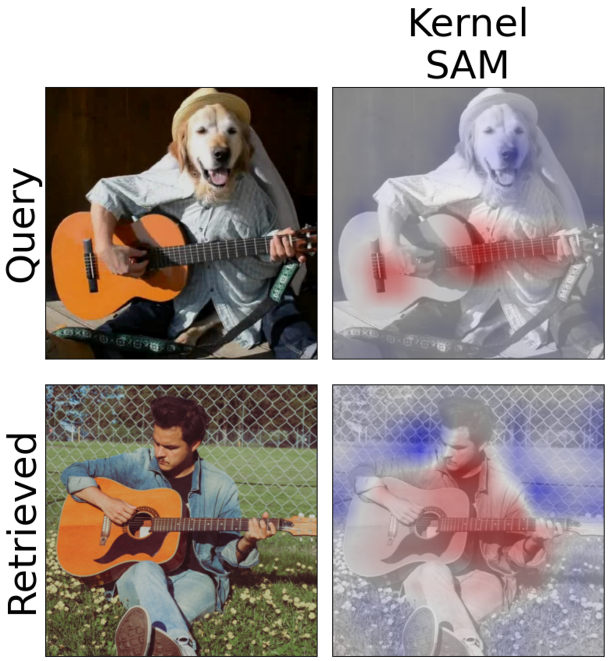


Figure E.5: Explanation of why two images are similar (Red) and dissimilar (Blue). Blue regions highlight major differences between the images such as the dog playing the guitar, and the chain-link fence in the retrieved image.

E.16 On the “Axiomatic” terminology

The term “axiomatic” can mean different things to different readers. When this work refers to “axiomatic” methods we refer to methods that approximate the uniquely specified explanation values dictated by the axioms of fair-credit assignment. In the first-order case, these explanations are the Shapey Values and satisfy the axioms of linearity, efficiency, dummy, and symmetry. In the higher-order case these fair credit assignments are the Shapley-Taylor Indices and satisfy analogous axioms [292]. We note that our methods *converge* to the true Shapley and Shapley-Taylor indices and thus the deviations that arise as part of convergence induce corresponding deviations from the axioms of fair credit assignment. Nevertheless, we find that these deviations become negligible as our methods converge to the true Shapley and Shapley-Taylor values. This starkly contrasts the behavior of methods that do not converge to values that satisfy the axioms of fair credit assignment such as GradCAM as shown in Figure E.2.

Appendix F

Appendix for Chapter 8

F.1 Additional Experiments on Debiasing Feature Learning

The following experiments aim to test the effect of our debiasing approach in feature learning. We followed the experimental setup introduced by Hu et al. [337]. The architecture consisted of a ResNet-34 backbone paired with a two-layer multilayer perceptron (MLP) feature extractor. The MLP included a hidden layer with 512 units and an output layer with 64 units, without batch normalization.

CIFAR-10 & CIFAR-100. The models were trained on the CIFAR-10 dataset for 1000 epochs using the AdamW optimizer with the following hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 1×10^{-3}, a batch size of 1024, and a weight decay of 1×10^{-5}. The learned kernel was either Gaussian or Student's t-distribution with degrees of freedom $d_f = 2$.

For evaluation, we used two methods: (1) linear probing on the 512-dimensional embeddings from the MLP's hidden layer, and (2) k-nearest neighbors ($k = 3$) classification based on the same embeddings for CIFAR-10 (in-distribution) and CIFAR-100 (out-of-distribution).

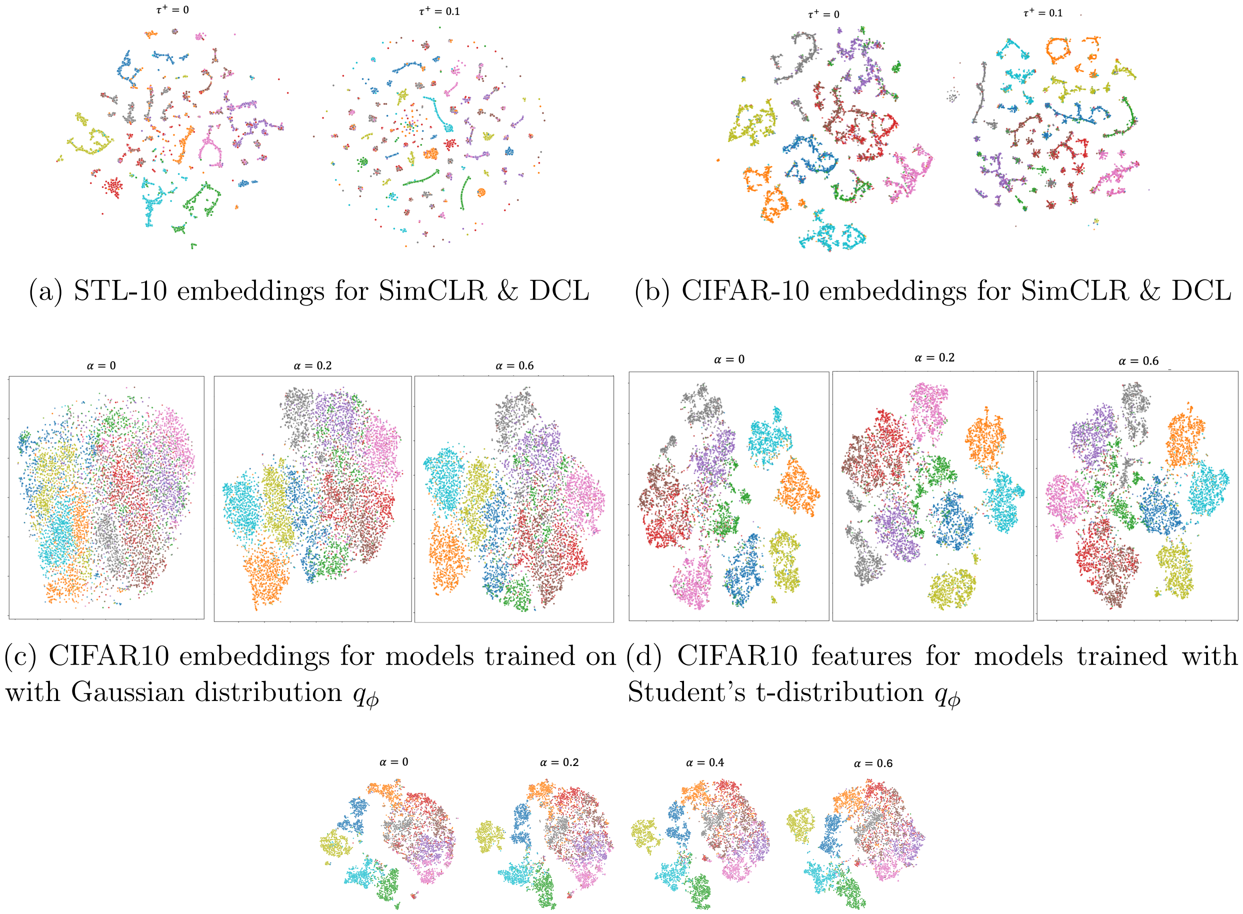
STL-10 & Oxford-IIIT Pet. With a similar setup, the models were trained contrastively on STL-10 (in distribution) without labels using the same hyperparameters as in the CIFAR experiments. For evaluation, we performed (1) linear probing for the STL-10 classification task and Oxford-IIIT Pet binary classification, and (2) k-nearest neighbors classification based on the same embeddings for STL-10 and Oxford-IIIT Pet with $k = 10$.

Method	**CIFAR10 (in distribution)**		**CIFAR100 (out of distribution)**	
	Linear Probing	KNN	Linear Probing	KNN
	q_ϕ is a Gaussian Distribution			
SimCLR [348]	77.79	80.02	31.82	40.27
DCL [369]	78.32	83.11	32.44	42.10
Ours $\alpha = 0.2$	79.50	84.07	32.53	43.19
Ours $\alpha = 0.4$	79.07	85.06	32.53	**43.29**
Ours $\alpha = 0.6$	79.32	85.90	30.67	29.79
	q_ϕ is a Student's t-distribution			
t-SimCLR[337]	90.97	88.14	38.96	30.75
DCL [369]	Diverges	Diverges	Diverges	Diverges
Ours $\alpha = 0.2$	91.31	88.34	41.62	32.88
Ours $\alpha = 0.4$	92.70	88.50	**41.98**	34.26
Ours $\alpha = 0.6$	**92.86**	**88.92**	38.92	32.51

Table F.1: Contrastive feature learning evaluation results for CIFAR10 and CIFAR100 datasets with various debasing α factors. Adding some amount of debasing helps raising accuracy in both linear probing and KNN classification.

Method	**STL-10 (in distribution)**		**Oxford-IIIT Pet (out of distribution)**	
	Linear Probing	KNN	Logistic Regression	KNN
SimCLR [348]	77.71	74.92	74.80	71.48
DCL [369]	78.32	75.03	74.41	70.22
q_ϕ is a Student's t-distribution				
t-SimCLR[337]	85.11	83.05	83.40	81.41
Ours $\alpha = 0.2$	85.94	83.15	84.11	81.15
Ours $\alpha = 0.4$	86.13	**84.14**	84.07	**84.13**
Ours $\alpha = 0.6$	**87.18**	83.58	**84.51**	83.04

Table F.2: Contrastive feature learning evaluation results for STL10 (in distribution) and Oxford-IIIT Pet (out of distribution) with various debasing α factors. Similar to the other experiments, our debasing helps raising accuracy in both linear probing and KNN classification.



(a) STL-10 embeddings for SimCLR & DCL

(b) CIFAR-10 embeddings for SimCLR & DCL

(c) CIFAR10 embeddings for models trained on with Gaussian distribution q_ϕ

(d) CIFAR10 features for models trained with Student's t-distribution q_ϕ

(e) STL-10 features for models trained with Student's t-distribution q_ϕ

Figure F.1: t-SNE visualizations of learned embeddings on CIFAR10 and STL10 datasets. (a) and (b) display embeddings from the DCL [369] method before and after applying debiasing, showing a tendency to heavily cluster data points, which may hinder out-of-distribution generalization [337]. (c) and (d) show embeddings with Student's t-distribution, where the debiasing factor α enhances clustering and separation, resulting in improved data representation.

F.2 Proofs for Unifying Dimensionality Reduction Methods

We begin by defining the setup for dimensionality reduction methods in the context of I-Con. Let $x_i \in \mathbb{R}^d$ represent high-dimensional data points, and $\phi_i \in \mathbb{R}^m$ represent their corresponding low-dimensional embeddings, where $m \ll d$. The goal of dimensionality reduction methods, such as Stochastic Neighbor Embedding (SNE) and t-Distributed Stochastic Neighbor Embedding (t-SNE), is to learn these embeddings such that neighborhood structures in the high-dimensional space are preserved in the low-dimensional space. In this context, the low-dimensional embeddings ϕ_i can be interpreted as the outputs of a mapping function $f_\theta(x_i)$, where f_θ is essentially an embedding matrix or look-up table. The I-Con framework is well-suited to express this relationship through a KL divergence loss between two neighborhood distributions: one in the high-dimensional space and one in the low-dimensional space.

Theorem F.2.1. *Stochastic Neighbor Embedding (SNE) [346] is an instance of the I-Con framework.*

Proof. This is one of the most straightforward proofs in this chapter, essentially based on the definition of SNE. The target distribution (supervised part), described by the neighborhood distribution in the high-dimensional space, is given by:

$$p_\theta(j|i) = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|^2/2\sigma_i^2\right)},$$

while the learned low-dimensional neighborhood distribution is:

$$q_\phi(j|i) = \frac{\exp\left(-\|\phi_i - \phi_j\|^2\right)}{\sum_{k\neq i}\exp\left(-\|\phi_i - \phi_k\|^2\right)}.$$

The objective is to minimize the KL divergence between these distributions:

$$\mathcal{L} = \sum_i D_{\mathrm{KL}}(p_\theta(\cdot|i)\|q_\phi(\cdot|i)) = \sum_i \sum_j p_\theta(j|i)\log\frac{p_\theta(j|i)}{q_\phi(j|i)}.$$

The embeddings θ_i are learned implicitly by minimizing $\mathcal{L}$. The mapper is an embedding matrix, as SNE is a non-parametric optimization. Therefore, SNE is a special case of the I-Con framework, where $p_\theta(j|i)$ and $q_\phi(j|i)$ represent the neighborhood probabilities in the high- and low-dimensional spaces, respectively. □

Corollary 1 (t-SNE [347])**.** *t-SNE is an instance of the I-Con framework.*

Proof. The proof is similar to the one for SNE. While the high-dimensional target distribution $p_\theta(j|i)$ remains unchanged, t-SNE modifies the low-dimensional distribution to a Student's t-distribution with one degree of freedom (Cauchy distribution):

$$q_\phi(j|i) = \frac{(1 + \|\phi_i - \phi_j\|^2)^{-1}}{\sum_{k\neq i}(1 + \|\phi_i - \phi_k\|^2)^{-1}}.$$

The objective remains to minimize the KL divergence. Therefore, t-SNE is an instance of the I-Con framework. □

Proposition F.2.1. *Let $X := \{x_i\}_{i=1}^n$, then the following cohesion variance loss*

$$\mathcal{L}_{\textit{cohesion-var}} = \frac{1}{n}\sum_{ij} w_{ij}\|f_\phi(x_i) - f_\phi(x_j)\|^2 - 2\,Var(X)$$

is an instance of $I - Con$ in the special case $w_{ij} = p(j|i)$ and q_ϕ is Gaussian as with a large width as $\sigma \to \infty$.

Proof. By using AM-GM inequality, we have

$$\frac{1}{n}\sum_{k=1}^{n} e^{-z_k} \geq (\Pi_{k=1}^{n} e^{-z_k})^{\frac{1}{n}} \implies \frac{1}{n}\sum_{k=1}^{n} e^{-z_k} \geq (e^{-\sum_{k=1}^{n} z_k})^{\frac{1}{n}}$$

which implies that

$$\log\sum_{k=1}^{n} e^{-z_k} - \log n \geq \log\left(e^{-\sum_{k=1}^{n} z_k}\right)^{\frac{1}{n}} \implies \log\sum_{k=1}^{n} e^{-z_k} \geq -\frac{1}{n}\sum_{k=1}^{n} z_k + \log(n)$$

Alternatively, this can be written as

$$-\log\sum_{k=1}^{n} e^{-z_k} \leq \frac{1}{n}\sum_{k=1}^{n} z_k - \log(n)$$

Now assume that we have a Gaussian Kernel q_ϕ

$$q_\phi(j|i) = \frac{\exp\left(-\|f_\phi(x_i) - f_\phi(x_j)\|^2/\sigma^2\right)}{\sum_{k\neq i}\exp\left(-\|f_\phi(x_i) - f_\phi(x_k)\|^2/\sigma^2\right)},$$

Therefore, given the inequality of exp-sum that we showed above, we have

$$\begin{aligned}
\log q_\phi(j|i) &= -\frac{\|f_\phi(x_i) - f_\phi(x_j)\|^2}{\sigma^2} - \log\sum_{k\neq i}\exp\left(-\frac{\|f_\phi(x_i) - f_\phi(x_k)\|^2}{\sigma^2}\right) \\
&\leq -\frac{1}{\sigma^2}\|f_\phi(x_i) - f_\phi(x_j)\|^2 + \frac{1}{n\sigma^2}\sum_{k\neq i}\|f_\phi(x_i) - f_\phi(x_k)\|^2 - \log(n) \\
&= -\frac{1}{\sigma^2}(-\|f_\phi(x_i) - f_\phi(x_j)\|^2 + \frac{1}{n}\sum_{k\neq i}\|f_\phi(x_i) - f_\phi(x_k)\|^2) - \log(n)
\end{aligned}$$

Therefore, the cross entropy $H(p_\theta, q_\phi)$, is bounded by

$$
\begin{aligned}
H(p_\theta, q_\phi) &= -\frac{1}{n}\sum_i\sum_j p(j|i)\log q_\phi(j|i) \\
&\leq \frac{1}{n}\sum_i\sum_j p(j|i)\left(\frac{1}{\sigma^2}(-\|f_\phi(x_i)-f_\phi(x_j)\|^2 + \frac{1}{n}\sum_{k\neq i}\|f_\phi(x_i)-f_\phi(x_k)\|^2) - \log(n)\right) \\
&= \frac{1}{\sigma^2}\left(\frac{1}{n}\sum_{ij} p(j|i)\|f_\phi(x_i)-f_\phi(x_j)\|^2 - \frac{1}{n^2}\sum_{ijk} p(j|i)\|f_\phi(x_i)-f_\phi(x_k)\|^2\right) - \log(n) \\
&= \frac{1}{\sigma^2}\left(\frac{1}{n}\sum_{ij} p(j|i)\|f_\phi(x_i)-f_\phi(x_j)\|^2 - 2\mathrm{Var}(X)\right) + \log(n) \\
&= \frac{1}{\sigma^2}\left(\frac{1}{n}\sum_{ij} p(j|i)\|f_\phi(x_i)-f_\phi(x_j)\|^2 - 2\mathrm{Var}(X)\right) + \log(n) \\
&= \frac{1}{\sigma^2}\mathcal{L}_{\text{cohesion-var}} + \log(n)
\end{aligned}
$$

On the other hand, the L.H.S. can be upper bounded by using second order bound $e^{-z} \leq 1 - z + z^2/2$, which implies that

$$
-\log\sum_{k=1}^n e^{-z_k} \geq \log(1 - \frac{1}{n}\sum_{k=1}^n z_k + \frac{1}{n}\sum_{k=1}^n z_k^2) - \log(n)
$$

On the other hand, $\log(1+u) \geq u - u^2/2$, therefore,

$$
-\log\sum_{k=1}^n e^{-z_k} \geq (1 - \frac{1}{n}\sum_{k=1}^n z_k + \frac{1}{n}\sum_{k=1}^n z_k^2) - \frac{1}{2}(1 - \frac{1}{n}\sum_{k=1}^n z_k + \frac{1}{n}\sum_{k=1}^n z_k^2)^2 - \log(n)
$$

Therefore, in the limit $\sigma \to \infty$, the bounds become tighter and the I-Con loss approaches the cohesion variance loss. □

Theorem F.2.2. *Principal Component Analysis (PCA) is an asymptotic instance of the I-Con.*

Proof. By using Proposition F.2.1. When $p_{j|i} = \mathbf{1}[i = j]$, we have the following expression for $\mathcal{L}$

$$
\begin{aligned}
\mathcal{L} &= \frac{1}{n}\sum_{ij} p_{j|i}\|f_\phi(x_i)-f_\phi(x_j)\|^2 - 2\mathrm{Var}(X) \\
&= \frac{1}{n}\sum_i \|f_\phi(x_i)-f_\phi(x_i)\|^2 - 2\mathrm{Var}(X) \\
&= -2\mathrm{Var}(X)
\end{aligned}
$$

Therefore, minimizing $\mathcal{L}$ is equivalent to maximizing the variance which is the equivalent of the PCA objective. Intuitivily, the KL divergence is asking $-\|f_\phi(x_i) - f_\phi(x_i)\|^2 = 0$ to be the maximum in comparison to $-\|f_\phi(x_i) - f_\phi(x_j)\|^2$ to match the supervisory indicator function, which implies the minimization of the sum of $-\|f_\phi(x_i) - f_\phi(x_j)\|^2$, which is maximizing the variance. If we restrict f_ϕ to be a linear projection map, then minimizing $\mathcal{L}$ would be equivalent to PCA. □

F.3 Proofs for Unifying Feature Learning Methods

We now extend the I-Con framework to feature learning methods commonly used in contrastive learning. Let $x_i \in \mathbb{R}^d$ be the input data points, and $f_\phi(x_i) \in \mathbb{R}^m$ be their learned feature embedding. In contrastive learning, the goal is to learn these embeddings such that similar data points (positive pairs) are close in the embedding space, while dissimilar points (negative pairs) are far apart. This setup can be expressed using a neighborhood distribution in the original space, where "neighbors" are defined not by proximity in Euclidean space, but by predefined relationships such as data augmentations or class membership. The learned embeddings $f_\phi(x_i)$ define a new distribution over neighbors, typically using a Gaussian kernel in the learned feature space. We show that InfoNCE is a natural instance of the I-Con framework, and many other methods, such as SupCon, CMC, and Cross Entropy, follow from this.

Theorem F.3.1 (InfoNCE [360]). *InfoNCE is an instance of the I-Con framework.*

Proof. InfoNCE aims to maximize the similarity between positive pairs while minimizing it for negative pairs in the learned feature space. In the I-Con framework, this can be interpreted as minimizing the divergence between two distributions: the neighborhood distribution in the original space and the learned distribution in the embedding space.

The neighborhood distribution $p_\theta(j|i)$ is uniform over the positive pairs, defined as:

$$p_\theta(j|i) = \begin{cases} \frac{1}{k} & \text{if } x_j \text{ is among the } k \text{ positive views of } x_i, \\ 0 & \text{otherwise.} \end{cases}$$

where k is the number of positive pairs for x_i.

The learned distribution $q_\phi(j|i)$ is based on the similarities between the embeddings $f_\phi(x_i)$ and $f_\phi(x_j)$, constrained to unit norm ($\|f_\phi(x_i)\| = 1$). Using a temperature-scaled Gaussian kernel, this distribution is given by:

$$q_\phi(j|i) = \frac{\exp\left(f_\phi(x_i) \cdot f_\phi(x_j)/\tau\right)}{\sum_{k \neq i} \exp\left(f_\phi(x_i) \cdot f_\phi(x_k)/\tau\right)},$$

where τ is the temperature parameter controlling the sharpness of the distribution. Since $\|f_\phi(x_i)\| = 1$, the Euclidean distance between $f_\phi(x_i)$ and $f_\phi(x_j)$ is $2 - 2(f_\phi(x_i) \cdot f_\phi(x_j))$.

The InfoNCE loss can be written in its standard form:

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_i \log \frac{\exp\left(f_\phi(x_i) \cdot f_\phi(x_i^+)/\tau\right)}{\sum_k \exp\left(f_\phi(x_i) \cdot f_\phi(x_k)/\tau\right)},$$

where j^+ is the index of a positive pair for i. Alternatively, in terms of cross-entropy, the loss becomes:

$$\mathcal{L}_{\text{InfoNCE}} \propto \sum_i \sum_j p_\theta(j|i) \log q_\phi(j|i) = H(p_\theta, q_\phi),$$

where $H(p_\theta, q_\phi)$ denotes the cross-entropy between the two distributions. Since $p_\theta(j|i)$ is fixed, minimizing the cross-entropy $H(p_\theta, q_\phi)$ is equivalent to minimizing the KL divergence $D_{\text{KL}}(p_\theta \| q_\phi)$. By aligning the learned distribution $q_\phi(j|i)$ with the target distribution $p_\theta(j|i)$, InfoNCE operates within the I-Con framework, where the neighborhood structure in the original space is preserved in the embedding space. Thus, InfoNCE is a direct instance of I-Con, optimizing the same divergence-based objective. □

Corollary 2. *t-SimCLR and t-SimCNE [337, 339] are instances of the I-Con framework.*

Given the proof of Theorem F.3.1, we can see that t-SimCLR is equivelant by having the same p_θ but q_ϕ would change from a Gaussian distribution over cosine similarity to a Student-T distribution over a Euclidean distance.

$$q_\phi(j|i) = \frac{\left(\|f_\phi(x_i) - f_\phi(x_j)\|^2/\tau\right)^{-1}}{\sum_{k \neq i} \left(\|f_\phi(x_i) - f_\phi(x_k)\|^2/\tau\right)^{-1}},$$

Theorem F.3.2. *VICReg [362] without a covariance term is an instance of the I-Con framework.*

Given Proposition F.2.1, we know that any loss in the cohesion variance form is an instance of I-Con:

$$\mathcal{L} = \frac{1}{n} \sum_{ij} p_{j|i} \|f_\phi(x_i) - f_\phi(x_j)\|^2 - 2\text{Var}(X)$$

If we choose $p_{j|i}$ to be an indicator over positive pairs, i and i^+, we obtain

$$\mathcal{L} = \frac{1}{n} \sum_i \|f_\phi(x_i) - f_\phi(x_{i^+})\|^2 - 2\text{Var}(X)$$

which is the VICReg loss without the covariance term and with an invariance-to-variance term ratio of 1:2. Observe that VICReg does not have negative pairs because it applies an equal repulsion force to all points. This is equivalent to taking $\sigma \to \infty$ in the conditional Gaussian distribution over the embeddings.

Theorem F.3.3 (Triplet Loss [361])**.** *Triplet Loss can be viewed as an instance of the I-Con framework with the following distributions $p_\theta(j|i)$ and $q_\phi(j|i)$:*

$$p_\theta(j|i) = \begin{cases} \frac{1}{k} & \textit{if } x_j \textit{ is among the } k \textit{ positive views of } x_i, \\ 0 & \textit{otherwise}, \end{cases}$$

$$q_\phi(j|i) = \frac{\exp\left(-\frac{\|f_\phi(x_i) - f_\phi(x_j)\|^2}{\sigma^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|f_\phi(x_i) - f_\phi(x_k)\|^2}{\sigma^2}\right)},$$

particularly in the special case where only two neighbors are considered: one positive view and one negative view.

Proof. The idea of this proof was first presented at [363] using Taylor Approximation; however, in this proof we present a stronger bounds for this result. For simplicity, we set $\sigma = 1$ (the general bounds for other σ values are provided at the end of the proof).

$$\mathcal{L} = -\frac{1}{N} \sum_i \sum_j q_\phi(j|i) \log \frac{\exp\left(-\|f_\phi(x_i) - f_\phi(x_j)\|^2\right)}{\sum_{k \neq i} \exp\left(-\|f_\phi(x_i) - f_\phi(x_k)\|^2\right)}.$$

In the special case where each anchor x_i has exactly one positive x_i^+ and one negative x_i^- example, the denominator simplifies to:

$$\sum_{k \neq i} \exp\left(-\|f_\phi(x_i) - f_\phi(x_k)\|^2\right) = \exp\left(-\|f_\phi(x_i) - f_\phi(x_i^+)\|^2\right) + \exp\left(-\|f_\phi(x_i) - f_\phi(x_i^-)\|^2\right).$$

Let $d_i^+ = \|f_\phi(x_i) - f_\phi(x_i^+)\|^2$ and $d_i^- = \|f_\phi(x_i) - f_\phi(x_i^-)\|^2$. Substituting these into the loss function, we obtain:

$$\begin{aligned} \mathcal{L} &= -\frac{1}{N} \sum_i \log \frac{\exp\left(-d_i^+\right)}{\exp\left(-d_i^+\right) + \exp\left(-d_i^-\right)} \\ &= -\frac{1}{N} \sum_i \log \left(\frac{1}{1 + \exp\left(d_i^- - d_i^+\right)}\right) \\ &= \frac{1}{N} \sum_i \log\left(1 + \exp\left(d_i^+ - d_i^-\right)\right). \end{aligned}$$

Recognizing that the expression inside the logarithm is the softplus function, we can leverage its well-known bounds:

$$\max(z, 0) \leq \log\left(1 + \exp(z)\right) \leq \max(z, 0) + \log(2).$$

By letting $z = d_i^+ - d_i^-$, we substitute into the bounds to obtain:

$$\frac{1}{N} \sum_i \max(d_i^+ - d_i^-, 0) \leq \mathcal{L} \leq \frac{1}{N} \sum_i \max(d_i^+ - d_i^-, 0) + \log(2),$$

where the left-hand side is the Triplet loss $\mathcal{L}_{\text{Triplet}} = \frac{1}{N} \sum_i \max(d_i^+ - d_i^-, 0)$. Therefore, we obtain the following bounds:

$$\mathcal{L} - \log(2) \leq \mathcal{L}_{\text{Triplet}} \leq \mathcal{L}.$$

For a general σ, the inequality bounds are as follows:

$$\mathcal{L}_\sigma - \sigma^2 \log(2) \leq \mathcal{L}_{\text{Triplet}} \leq \mathcal{L}_\sigma,$$

where

$$\mathcal{L}_\sigma = -\frac{\sigma^2}{N} \sum_i \sum_j q_\phi(j|i) \log \frac{\exp\left(-\frac{\|f_\phi(x_i) - f_\phi(x_j)\|^2}{\sigma^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|f_\phi(x_i) - f_\phi(x_k)\|^2}{\sigma^2}\right)}.$$

As σ approaches 0, $\mathcal{L}_{\text{Triplet}}$ approaches $\mathcal{L}_\sigma$. □

Theorem F.3.4. *The Supervised Contrastive Loss [363] is an instance of the I-Con framework.*

Proof. This follows directly from Theorem F.3.1. Define the supervisory and target distributions as:

$$q_\phi(j \mid i) = \frac{\exp\left(f_\phi(x_i) \cdot f_\phi(x_j)/\tau\right)}{\sum_{k \neq i} \exp\left(f_\phi(x_i) \cdot f_\phi(x_k)/\tau\right)},$$

$$p_\theta(j \mid i) = \frac{1}{K_i - 1}\mathbf{1}[i \text{ and } j \text{ share the same label}],$$

where f_ϕ is the mapping to deep feature space and K_i is the number of samples in the class of i. Substituting these definitions into the I-Con framework recovers the Supervised Contrastive Loss. □

Theorem F.3.5. *The X-Sample Contrastive Learning Loss [336] is an instance of the I-Con framework.*

Proof. Consier the following p distribution over corresponding features (e.g. caption embeddings for images):

$$\frac{\exp\big(g_\theta(x_i) \cdot g_\theta(x_j)\big)}{\sum\limits_{k \neq i} \exp\big(g_\theta(x_i) \cdot_\theta (x_k)\big)}$$

where g could be either a parametric or a non-parametric mapper to the corresponding embeddings $g_\theta(x_i)$. On the other hand, similar to most feature learning methods, the learned distribution is a Gaussian over learned embeddings with cosine distance

$$q_\phi(j \mid i) = \frac{\exp\big(f_\phi(x_i) \cdot f_\phi(x_j)\big)}{\sum\limits_{k \neq i} \exp\big(f_\phi(x_i) \cdot f_\phi(x_k)\big)}$$

where f_ϕ is the mapping to deep feature space. □

Theorem F.3.6. *Contrastive Multiview Coding (CMC) and CLIP are instances of the I-Con framework.*

Proof. Since we have already established that InfoNCE is an instance of the I-Con framework, this corollary follows naturally. The key difference in Contrastive Multiview Coding (CMC) and CLIP is that they optimize alignment across different modalities. The target probability distribution $p_\theta(j|i)$ can be expressed as:

$$p_\theta(j|i) = \frac{1}{Z}\mathbb{1}[i \text{ and } j \text{ are positive pairs and } V_i \neq V_j],$$

where V_i and V_j represent the modality sets of x_i and x_j, respectively. Here, $p_\theta(j|i)$ assigns uniform probability over positive pairs drawn from different modalities.

The learned distribution $q_\phi(j|i)$, in this case, is based on a Gaussian similarity between deep features, but conditioned on points from the opposite modality set. Thus, the learned distribution is defined as:

$$q_\phi(j|i) = \frac{\exp\left(-\|f_\phi(x_i) - f_\phi(x_j)\|^2\right)}{\sum_{k \in V_j} \exp\left(-\|f_\phi(x_i) - f_\phi(x_k)\|^2\right)}.$$

This formulation shows that CMC and CLIP follow the same principles as InfoNCE but apply them in a multiview setting, fitting seamlessly within the I-Con framework by minimizing the divergence between the target and learned distributions across different modalities. □

Theorem F.3.7. *Cross-Entropy classification is an instance of the I-Con framework.*

Proof. Cross-Entropy can be viewed as a special case of the CMC loss, where one "view" corresponds to the data point features and the other to the class logits. The affinity between a data point and a class is based on whether the point belongs to that class. This interpretation has been explored in prior work, where Cross-Entropy was shown to be related to the CLIP loss [338]. □

Theorem F.3.8. *Harmonic Loss for classification is an instance of the I-Con framework.*

Proof. This is the equivalent of moving from a Gaussian distribution for $q(j|i)$ in Cross-Entropy to a Student-T distribution analogs to moving from SNE to t-SNE. More specifically, let V be the set of data points, C the set of class prototypes, ϕ_i be the learned class prototype for class i, and n be the harmonic loss degree.

Consider the following p, which is a data-label indicator

$$p(j|i) = \mathbb{1}\left[i \text{ belongs to class } j\right]$$

and the following q, which is a Student-T distribution with $2n-1$ degrees for freedom.

$$\lim_{\sigma\to 0} \frac{(1+\|f_\phi(x_i)-\phi_j\|^2/((2n-1)\sigma^2))^{-n}}{\sum_{k\in C}(1+\|f_\phi(x_i)-\phi_k\|^2/((2n-1)\sigma^2))^{-n}}$$

It can be rewritten as

$$\lim_{\sigma\to 0} \frac{(((2n-1)\sigma^2)+\|f_\phi(x_i)-\phi_j\|^2)^{-n}}{\sum_{k\in C}(((2n-1)\sigma^2)+\|f_\phi(x_i)-\phi_k\|^2/)^{-n}}$$

As $\sigma \to \infty$, the loss function approaches

$$\mathcal{L} = \sum_{i\in C} \frac{(\|f_\phi(x_i)-\phi_j\|^2)^{-n}}{\sum_{k\in C}(\|f_\phi(x_i)-\phi_k\|^2/)^{-n}}$$

which's the Harmonic Loss for classification as introduced by F.3.8 □

Theorem F.3.9. *Masked Language Modeling (MLM) [367] loss is an instance of the I-Con framework.*

Proof. In Masked Language Modeling, the objective is to predict a masked token j given its surrounding context x_i. This setup fits naturally within the I-Con framework by defining appropriate target and learned distributions.

The target distribution $p_\theta(j|i)$ is the empirical distribution over contexts i and tokens j, defined as:

$$p_\theta(j|i) = \frac{1}{Z}\#\left[\text{Context } i \text{ precedes token } j\right],$$

where $\#\left[\text{Context } i \text{ precedes token } j\right]$ counts the number of times token j follows context x_i in the training corpus and Z is a normalization constant ensuring that $\sum_j p_\theta(j|i) = 1$.

The learned distribution $q_\phi(j|i)$ is modeled using the neural network's output logits for token predictions. It is defined as a softmax over the dot product between the context embedding $f_\phi(x_i)$ and the token embeddings ϕ_j:

$$q_\phi(j|i) = \frac{\exp\left(f_\phi(x_i) \cdot \phi_j\right)}{\sum_{k \in \mathcal{V}} \exp\left(f_\phi(x_i) \cdot \phi_k\right)},$$

where $f_\phi(x_i)$ is the embedding of the context x_i produced by the model, ϕ_j is the embedding of token j, and $\mathcal{V}$ is the vocabulary of all possible tokens.

The MLM loss aims to minimize the cross-entropy between the target distribution $p_\theta(j|i)$ and the learned distribution $q_\phi(j|i)$:

$$\mathcal{L}_{\text{MLM}} = -\sum_i \sum_j p_\theta(j|i) \log q_\phi(j|i) = H(p_\theta, q_\phi).$$

Since in practice, for each context x_i, only the true masked token j_i^* is considered, the target distribution simplifies to:

$$p_\theta(j|i) = \delta_{j, j_i^*},$$

where δ_{j,j_i^*} is the Kronecker delta function, equal to 1 if $j = j_i^*$ and 0 otherwise.

Substituting this into the loss function, the MLM loss becomes:

$$\mathcal{L}_{\text{MLM}} = -\sum_i \log q_\phi(j_i^* | x_i).$$

□

F.4 Proofs for Unifying Clustering Methods

The connections between clustering and the I-Con framework are more intricate compared to the dimensionality reduction methods discussed earlier. To establish these links, we first introduce a probabilistic formulation of K-means and demonstrate its equivalence to the classical K-means algorithm, showing that it is a zero-gap relaxation. Building upon this, we reveal how probabilistic K-means can be viewed as an instance of I-Con, leading to a novel clustering kernel. Finally, we show that several clustering methods implicitly approximate and optimize for this kernel.

Definition 1 (Classical K-means)**.** *Let $x_1, x_2, \ldots, x_N \in \mathbb{R}^n$ denote the data points, and $\mu_1, \mu_2, \ldots, \mu_m \in \mathbb{R}^n$ be the cluster centers.*

The objective of classical K-means is to minimize the following loss function:

$$\mathcal{L}_{\textit{k-Means}} = \sum_{i=1}^{N} \sum_{c=1}^{m} \mathbb{1}(c^{(i)} = c) \|x_i - \mu_c\|^2,$$

where $c^{(i)}$ represents the cluster assignment for data point x_i, and is defined as:

$$c^{(i)} = \arg\min_c \|x_i - \mu_c\|^2.$$

Probabilistic K-means Relaxation

In probabilistic K-means, the cluster assignments are relaxed by assuming that each data point x_i belongs to a cluster c with probability ϕ_{ic}. In other words, ϕ_i represents the cluster assignments vector for x_i

Proposition F.4.1. *The relaxed loss function for probabilistic K-means is given by:*

$$\mathcal{L}_{\textit{Prob-k-Means}} = \sum_{i=1}^{N} \sum_{c=1}^{m} \phi_{ic} \|x_i - \mu_c\|^2,$$

and is equivalent to the original K-means objective $\mathcal{L}_{\textit{k-Means}}$. The optimal assignment probabilities ϕ_{ic} are deterministic, assigning probability 1 to the closest cluster and 0 to others.

Proof. For each data point x_i, the term $\sum_{c=1}^{m} \phi_{ic} \|x_i - \mu_c\|^2$ is minimized when the assignment probabilities ϕ_{ic} are deterministic, i.e.,

$$\phi_{ic} = \begin{cases} 1 & \text{if } c = \arg\min_j \|x_i - \mu_j\|^2, \\ 0 & \text{otherwise.} \end{cases}$$

With these deterministic probabilities, $\mathcal{L}_{\text{Prob-k-Means}}$ simplifies to the classical K-means objective, confirming that the relaxation introduces no gap. □

Contrastive Formulation of Probabilistic K-means

Definition 2. *Let $\{x_i\}_{i=1}^{N}$ be a set of data points. Define the conditional probablity $q_\phi(j|i)$ as:*

$$q_\phi(j|i) = \sum_{c=1}^{m} \frac{\phi_{ic}\phi_{jc}}{\sum_{k=1}^{N} \phi_{kc}},$$

where ϕ_i is the soft-cluster assignments for x_i.

Given $q_\phi(j|i)$, we can reformulate probabilistic K-means as a contrastive loss:

Theorem F.4.1. *Let $\{x_i\}_{i=1}^{N} \in \mathbb{R}^n$ and $\{\phi_{ic}\}_{i=1}^{N}$ be the corresponding assignment probabilities. Define the objective function $\mathcal{L}$ as:*

$$\mathcal{L} = -\sum_{i,j} (x_i \cdot x_j) \, q_\phi(j|i).$$

Minimizing $\mathcal{L}$ with respect to the assignment probabilities $\{\phi_{ic}\}$ yields optimal cluster assignments equivalent to those obtained by K-means.

Proof. The relaxed probabilistic K-means objective $\mathcal{L}_{\text{Prob-k-Means}}$ is:

$$\mathcal{L}_{\text{Prob-k-Means}} = \sum_{i=1}^{N} \sum_{c=1}^{m} \phi_{ic} \|x_i - \mu_c\|^2.$$

Expanding this, we obtain:

$$\mathcal{L}_{\text{Prob-k-Means}} = \sum_{c=1}^{m} \left(\sum_{i=1}^{N} \phi_{ic} \right) \|\mu_c\|^2 - 2 \sum_{c=1}^{m} \left(\sum_{i=1}^{N} \phi_{ic} x_i \right) \cdot \mu_c + \sum_{i=1}^{N} \|x_i\|^2.$$

The cluster centers μ_c that minimize this loss are given by:

$$\mu_c = \frac{\sum_{i=1}^{N} \phi_{ic} x_i}{\sum_{i=1}^{N} \phi_{ic}}.$$

Substituting μ_c back into the loss function, we get:

$$\mathcal{L} = -\sum_{i,j} (x_i \cdot x_j)\, q_\phi(j|i),$$

which proves that minimizing this contrastive formulation leads to the same clustering assignments as classical K-means. □

Corollary 3. *The alternative loss function:*

$$\mathcal{L} = -\sum_{i,j} \|x_i - x_j\|^2 \, q_\phi(j|i),$$

yields the same optimal clustering assignments when minimized with respect to $\{\phi_{ic}\}$.

Proof. Expanding the squared norm in the loss function gives:

$$\mathcal{L} = -\sum_{i,j} \left(\|x_i\|^2 - 2x_i \cdot x_j + \|x_j\|^2 \right) q_\phi(j|i).$$

The terms involving $\|x_i\|^2$ and $\|x_j\|^2$ simplify since $\sum_j q_\phi(j|i) = 1$, reducing the loss to:

$$\mathcal{L} = 2 \left(-\sum_{i,j} x_i \cdot x_j q_\phi(j|i) \right),$$

which is equivalent to the objective in the previous theorem. □

Probabilistic K-means as an I-Con Method

In the I-Con framework, the target and learned distributions represent affinities between data points based on specific measures. For instance, in SNE, these measures are Euclidean distances in high- and low-dimensional spaces, while in SupCon, the distances reflect whether data points belong to the same class. Similarly, we can define a measure of neighborhood probabilities in the context of clustering, where two points are considered neighbors if they belong to the same cluster. The probability of selecting x_j as x_i's neighbor is the probability that a point, chosen uniformly at random from x_i's cluster, is x_j. More explicitly, let $q_\phi(j|i)$ represent the probability that x_j is selected uniformly at random from x_i's cluster:

$$q_\phi(j|i) = \sum_{c=1}^{m} \frac{\phi_{ic}\phi_{jc}}{\sum_{k=1}^{N} \phi_{kc}}.$$

Theorem F.4.2 (K-means as an instance of I-Con)**.** *Given data points* $\{x_i\}_{i=1}^N$*, define the neighborhood probabilities* $p_\theta(j|i)$ *and* $q_\phi(j|i)$ *as:*

$$p_\theta(j|i) = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma^2\right)}{\sum_k \exp\left(-\|x_i - x_k\|^2/2\sigma^2\right)}, \quad q_\phi(j|i) = \sum_{c=1}^{m} \frac{\phi_{ic}\phi_{jc}}{\sum_{k=1}^N \phi_{kc}}.$$

Let the loss function $\mathcal{L}_{\textit{c-SNE}}$ *be the sum of KL divergences between the distributions* $q_\phi(j|i)$ *and* $p_\theta(j|i)$*:*

$$\mathcal{L}_{\textit{c-SNE}} = \sum_i D_{KL}(q_\phi(\cdot|i)\|p_\theta(\cdot|i)).$$

Then,

$$\mathcal{L}_{\textit{c-SNE}} = \frac{1}{2\sigma^2}\mathcal{L}_{\textit{Prob-k-Means}} - \sum_i H(q_\phi(\cdot|i)),$$

where $H(q_\phi(\cdot|i))$ *is the entropy of* $q_\phi(\cdot|i)$*.*

Proof. For simplicity, assume that $2\sigma^2 = 1$. Denote $\sum_k \exp\left(-\|x_i - x_k\|^2\right)$ by Z_i. Then we have:

$$\log p_\theta(j|i) = -\|x_i - x_j\|^2 - \log Z_i.$$

Let $\mathcal{L}_i$ be defined as $-\sum_j \|x_i - x_j\|^2 q_\phi(j|i)$. Using the equation above, $\mathcal{L}_i$ can be rewritten as:

$$\begin{aligned}
\mathcal{L}_i &= -\sum_j \|x_i - x_j\|^2 q_\phi(j|i) && \text{(F.1)}\\
&= \sum_j (\log(p_\theta(j|i)) + \log(Z_i)) q_\phi(j|i) && \text{(F.2)}\\
&= \sum_j q_\phi(j|i)\log(p_\theta(j|i)) + \sum_j q_\phi(j|i)\log(Z_i) && \text{(F.3)}\\
&= \sum_j q_\phi(j|i)\log(p_\theta(j|i)) + \log(Z_i) && \text{(F.4)}\\
&= H(q_\phi(\cdot|i), p_\theta(\cdot|i)) + \log(Z_i) && \text{(F.5)}\\
&= D_{\mathrm{KL}}(q_\phi(\cdot|i)\|p_\theta(\cdot|i)) + H(q_\phi(\cdot|i)) + \log(Z_i). && \text{(F.6)}
\end{aligned}$$

Therefore, $\mathcal{L}_{\text{Prob-KMeans}}$, as defined in Corollary 3, can be rewritten as:

$$\begin{aligned}
\mathcal{L}_{\text{Prob-KMeans}} &= -\sum_i \sum_j \|x_i - x_j\|^2 q_\phi(j|i) = \sum_i \mathcal{L}_i && \text{(F.7)}\\
&= \sum_i D_{\mathrm{KL}}(q_\phi(\cdot|i)\|p_\theta(\cdot|i)) + H(q_\phi(\cdot|i)) + \log(Z_i) && \text{(F.8)}\\
&= \mathcal{L}_{\text{c-SNE}} + \sum_i H(q_\phi(\cdot|i)) + \text{constant}. && \text{(F.9)}
\end{aligned}$$

Therefore,

$$\mathcal{L}_{\text{c-SNE}} = \mathcal{L}_{\text{Prob-KMeans}} - \sum_i H(q_\phi(\cdot|i)).$$

If we allow σ to take any value, the entropy penalty will be weighted accordingly:

$$\mathcal{L}_{\text{c-SNE}} = \frac{1}{2\sigma^2}\mathcal{L}_{\text{Prob-KMeans}} - \sum_i H(q_\phi(\cdot|i)).$$

Note that the relation above is up to an additive constant. This implies that minimizing the contrastive probabilistic K-means loss with entropy regularization minimizes the sum of KL divergences between $q_\phi(\cdot|i)$ and $p_\theta(\cdot|i)$. □

Corollary 4. *Spectral Clustering is an instance of the I-Con framework.*

Proof. From Theorem F.4.2, we know that K-Means clustering can be formulated as an instance of the I-Con framework, where the clustering assignments depend on the inner products of the data points.

Spectral Clustering extends this idea by first embedding the data into a lower-dimensional space using the top k eigenvectors of the normalized Laplacian derived from the affinity matrix A. The affinity matrix A is constructed using a similarity measure (e.g., an RBF kernel) and encodes the probabilities of assignments between data points. Given this transformation, spectral clustering is an instance of I-Con on the projected embeddings. □

Theorem F.4.3. *Normalized Cuts [354] is an instance of I-Con.*

Proof. The proof for this follows naturally from our work on K-Means analysis. The loss function for normalized cuts is defined as:

$$\mathcal{L}_{\text{NormCuts}} = \sum_{c=1}^{m} \frac{\text{cut}(A_c, \overline{A}_c)}{\text{vol}(A_c)},$$

where A_c is a subset of the data corresponding to cluster c, $\overline{A}_c$ is its complement, and $\text{cut}(A_c, \overline{A}_c)$ represents the sum of edge weights between A_c and $\overline{A}_c$, while $\text{vol}(A_c)$ is the total volume of cluster A_c, i.e., the sum of edge weights within A_c.

Similar to K-Means, by reformulating this in a contrastive style with soft-assignments, the learned distribution can be expressed using the probabilistic cluster assignments $\phi_{ic} = p(c|x_i)$ as:

$$q_\phi(j|i) = \sum_{c=1}^{m} \frac{\phi_{ic}\phi_{jc}d_j}{\sum_{k=1}^{N} \phi_{kc}d_k},$$

where d_j is the degree of node x_j, and the volume and cut terms can be viewed as weighted sums over the soft-assignments of data points to clusters.

This reformulation shows that normalized cuts can be written in a manner consistent with the I-Con framework, where the target distribution $p_\theta(j|i)$ and the learned distribution $q_\phi(j|i)$ represent affinity relationships based on graph structure and cluster assignments.

Thus, normalized cuts is an instance of I-Con, where the loss function optimizes the neighborhood structure based on the cut and volume of clusters in a manner similar to K-Means and its probabilistic relaxations. □

Theorem F.4.4. *Mutual Information Clustering is an instance of I-Con.*

Proof. Given the connection established between SimCLR, K-Means, and the I-Con framework, this result follows naturally. Specifically, the target distribution $p_\theta(j|i)$ (the supervised part) is a uniform distribution over observed positive pairs:

$$p_\theta(j|i) = \begin{cases} \frac{1}{k} & \text{if } x_j \text{ is among the } k \text{ positive views of } x_i, \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, the learned embeddings ϕ_i represent the probabilistic assignments of x_i into clusters. Therefore, similar to the analysis of the K-Means connection, the learned distribution is modeled as:

$$q_\phi(j|i) = \sum_{c=1}^{m} \frac{\phi_{ic}\phi_{jc}}{\sum_{k=1}^{N} \phi_{kc}}.$$

This shows that Mutual Information Clustering can be viewed as a method within the I-Con framework, where the learned distribution $q_\phi(j|i)$ aligns with the target distribution $p_\theta(j|i)$, completing the proof. □

F.5 I-Con as a Variational Method

Variational bounds for mutual information are widely explored and have been connected to loss functions such as InfoNCE, where minimizing InfoNCE maximizes the mutual information lower bound [101, 401]. The proof usually starts by rewriting the mutual information:

$$I(X;Y) = \mathbb{E}_{p(x,y)}\left[\log \frac{q(x|y)}{p(x)}\right] + \mathbb{E}_{p(y)}\left[D_{\mathrm{KL}}\left(p(x|y) \,\|\, q(x|y)\right)\right]$$

This expression is typically used to derive a lower bound for $I(X;Y)$. The proof usually begins by assuming that p is uniform over discrete data points $\mathcal{X} = \{x_i\}_{i=1}^N$ (i.e., we use uniform sampling for data points). By using the fact that $p(x_i) = \frac{1}{N}$, we can write $p(x,y) = \frac{1}{N}p(x|y)$. Therefore, the mutual information lower bound becomes

$$\begin{aligned} I(X;Y) &\geq \mathbb{E}_{p(x,y)}\left[\log q(x|y)\right] - \mathbb{E}_{p(x,y)}\left[\log p(x)\right] \\ &= \mathbb{E}_{p(x,y)}\left[\log q(x|y)\right] + \log(N) \\ &= \frac{1}{N} \sum_{x,y \in \mathcal{X}\times\mathcal{X}} p(x|y) \log q(x|y) + \log(N) \\ &= \frac{1}{N} \sum_{y \in \mathcal{X}} \sum_{x \in \mathcal{X}} p(x|y) \log q(x|y) + \log(N) \\ &= -H\left(p(x|y), q(x|y)\right) + \log(N) \end{aligned}$$

Therefore, maximizing the cross-entropy between the two distributions maximizes the mutual information between samples.

On the hand, Variational Bayesian (VB) methods are fundamental in approximating intractable posterior distributions $p(z \mid x)$ with tractable variational distributions $q_\phi(z)$. This approximation is achieved by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior:

$$\mathrm{KL}(q_\phi(z)\|p(z \mid x)) = \mathbb{E}_{q_\phi(z)}\left[\log \frac{q_\phi(z)}{p(z \mid x)}\right]. \tag{F.10}$$

The optimization objective, known as the Evidence Lower Bound (ELBO), is given by:

$$\mathrm{ELBO} = \mathbb{E}_{q_\phi(z)}\left[\log p(x, z)\right] - \mathbb{E}_{q_\phi(z)}\left[\log q_\phi(z)\right]. \tag{F.11}$$

Maximizing the ELBO is equivalent to minimizing the KL divergence, thereby ensuring that $q_\phi(z)$ closely approximates $p(z \mid x)$ [402].

VB can be framed within the I-Con framework by making specific mappings between the variables and distributions. Let i correspond to the data point x, and j correspond to the latent variable z. We can set the supervisory distribution $p_\theta(z \mid x)$ to be the true posterior $p(z \mid x)$. This allow us to define the learned distribution $q_\phi(z \mid x)$ to be independent of x, i.e., $q_\phi(z \mid x) = q_\phi(z)$.

Under these settings, the I-Con loss simplifies to:

$$\mathcal{L}(\phi) = \int_{x \in \mathcal{X}} \mathrm{KL}\left(p(z \mid x)\|q_\phi(z)\right)\, dx = \mathbb{E}_{p(x)}\left[\mathrm{KL}(p(z \mid x)\|q_\phi(z))\right]. \tag{F.12}$$

Interpretation

- Global Approximation: In VB, $q_\phi(z)$ serves as a global approximation to the posterior $p(z \mid x)$ across all data points x. Similarly, in I-Con, when $q_\phi(j \mid i) = q_\phi(j)$, the learned distribution provides a uniform approximation across all i.

- Variational Alignment: Both frameworks employ variational techniques to align a tractable distribution q_ϕ with an intractable or supervisory distribution p. This alignment ensures that the learned representations capture essential information from the target distribution.

- Framework Generalization: I-Con generalizes VB by allowing $q_\phi(j \mid i)$ to depend on i, enabling more flexible and data-specific alignments. VB is recovered as a special case where the learned distribution is uniform across all data points.

F.6 Why do we need to unify representation learners?

I-con not only provides a deeper understanding of these methods but also opens up the possibility of creating new methods by mixing and matching components. We explicitly use this property to discover new improvements to both clustering and representation learners. In short, I-Con acts like a periodic table of machine learning losses. With this periodic table we can more clearly see the implicit assumptions of each method by breaking down modern ML losses into more simple components: pairwise conditional distributions p and q.

One particular example of how this opens new possibilities is with our generalized debiasing operation. Through our experiments we show adding a slight constant linkage between datapoints improves both stability and performance across clustering and feature learning. Unlike prior art, which only applies to specific feature learners, our debiasers can improve clusterers, feature learners, spectral graph methods, and dimensionality reducers.

Finally it allows us to discover novel theoretical connections by compositionally exploring the space, and considering limiting conditions. We use I-Con to help derive a novel theoretical equivalences between K-Means and contrastive learning, and between MDS, PCA, and SNE. Transferring ideas between methods is standard in research, but in our view it becomes much simpler to do this if you know methods are equivalent. Previously, it might not be clear how exactly to translate an insight like changing Gaussian distributions to Cauchy distributions in the upgrade from SNE to T-SNE has any effect on clustering or representation learning. In I-Con it becomes clear to see that similarly softening clustering and representation learning distributions can improve performance and debias representations.

F.7 How to choose neighborhood distributions for your problem

Parameterization of Learning Signal

- **Parametric**: (Learn a network to transform a data points to representations). Use a parametric method to quickly represent new datapoints without retraining. Use a parametric method if there is enough "features" in the underlying data to properly learn a representation. Use this option with datasets with sparse supervisory signal in order to share learning signal through network parameters.

- **Nonparametric**: (Learn one representation per data point). Use a nonparametric method if datapoints are abstract and don't contain natural features that are useful for mapping. Use this option to better optimize the loss of each individual datapoint. Do not use this in sparse supervisory signal regimes (Like augmentation based contrastive learning), as there are not enough links to resolve each individual embedding.

Choice of supervisory signal

- **Gaussians on distances in the input space**: though this is a common choice and underlies methods like k-means, with enough data it is almost always better to use k-neighbor distributions as they better capture local topology of data. This is the same intuition that is used to justify spectral clustering over k-means.

- **K-neighbor graphs distributions**: If your data can be naturally put into a graph instead of just considering Gaussians on the input space we suggest it. This allows the algorithm to adapt local neighborhoods to the data, as opposed to considering all points neighborhoods equally shaped and sized. This better aligns with the manifold hypothesis.

- **Contrastive augmentations**: When possible, add contrastive augmentations to your graph - this will improve performance in cases where quantities of interest (like an image class) are guaranteed to be shared between augmentations.

- **General kernel smoothing techniques**: Use random walks to improve the optimization quality. It connects more points together and in some cases mirrors geodesic distance on the manifold [403].
- **Debiasing**: Use this if you think negative pairs actually have a small chance of aligning positively. For a small number of classes this parameter scales like the inverse of the number of classes. You can also use this to improve stability of the optimization.

Choice of representation:

Any conditional distribution on representations can be used, so consider what kind of structure you want to learn, tree, vector, cluster, etc. And choose the distribution to be simple and meaningful for that representation.

- **Discrete**: Use discrete cluster-based representations if interpretability and discrete structure are important
- **Continuous Vector**: Use a vector representation if generic downstream performance is a concern as this is a bit easier to optimize than discrete variants.

F.8 Comparing I-Con, MLE, and the KL Divergence

There are many connections between KL divergence and maximum likelihood estimation. We highlight the differences between a standard MLE approach and I-Con. In short, although I-Con has a maximum likelihood interpretation, its specific functional form allows it to unify both unsupervised and supervised methods in a way that elucidates the key structures that are important for deriving new representation learning losses. This is in contrast to the commonly known connection between MLE and KL divergence minimization, which does not focus on pairwise connections between datapoints and does not provide as much insight for representation learners. To see this we note that the conventional connection between MLE and KL minimization is as follows:

$$\theta_{\text{MLE}} = \arg\min_{\theta} D_{\text{KL}}(\hat{P}||Q_\theta),$$

where the empirical distribution, $\hat{P}$, is defined as:

$$\hat{P}(x) = \frac{1}{N}\sum_{i=1}^{N} \delta(x - x_i),$$

where $\delta(x-x_i)$ is the Dirac delta function. The classical KL minimization fits a parameterized model family to an empirical distribution. In contrast the I-Con equation:

$$\mathcal{L}(\theta, \phi) = \int_{i \in \mathcal{X}} D_{\text{KL}}\left(p_\theta(\cdot|i)||q_\phi(\cdot|i)\right)$$

Operates on conditional distributions and captures an "average" KL divergence instead of a single KL divergence. Secondly, I-Con explicitly involves a computation over neighboring datapoints which does not appear in the aforementioned equation. This decomposition of methods into their actions on their neighborhoods makes many methods simpler to understand, and makes modifications of these methods easier to transfer between domains. It also makes it possible to apply this theory to unsupervised problems where empirical supervisory data does not exist. Furthermore some methods, like DINO, do not share the exact functional form of I-Con, and suffer from various difficulties like collapse which need to be handled with specific regularizers. This shows that I-Con is not just a catchall reformulation of MLE, but is capturing a specific functional form shared by several popular learners.

F.9 On I-Con's Hyperparameters

One important way that I-Con removes hyperparameters from existing works is that it does not rely on things like entropy penalties, activation normalization, activation sharpening, or EMA stabilization to avoid collapse. The loss is self-balancing in this regard as any way that it can improve the learned distribution to better match the target distribution is "fair game". This allows one to generalize certain aspects of existing losses like InfoNCE. In I-Con info NCE looks like fixed-width Gaussian kernels mediating similarity between representation vectors. In I-Con it's trivial to generalize these Gaussians to have adaptive and learned covariances for example. This allows the network to select its own level of certainty in representation learning. If you did this naively, you would need to ensure the loss function doesn't cheat by making everything less certain.

Nevertheless I-Con defines a space of methods depending on the choice of p and q. The choice of these two distributions becomes the main source of hyperparameters we explore. In particular our experiments change the structure of the supervisory signal (often p). For example, in a clustering experiment changing p from "Gaussians with respect to distance" to "graph adjacency" transforms K-Means into Spectral clustering. It's important to note that K-means has benefits over Spectral clustering in certain circumstances and vice-versa, and there's not necessarily a singular "right" choice for p in every problem. Like many things in ML, the different supervisory distributions provide different inductive biases and should be chosen thoughtfully. We find that this design space makes it easier to build better performing supervisory signals for specific important problems like unsupervised image classification on ImageNet and others.

References

[1] C. Liu, M. Amodio, L. Shen, F. Gao, A. Avesta, S. Aneja, J. C. Wang, L. V. D. Priore, and S. Krishnaswamy. "Cuts: A Framework for Multigranular Unsupervised Medical Image Segmentation". In: *MICCAI*. 2024.

[2] Z. Chen, H. Xu, W. Chen, Z. Zhou, H. Xiao, B. Sun, and X. Xie. "PointDC: Unsupervised Semantic Segmentation of 3D Point Clouds via Cross-modal Distillation and Super-Voxel Clustering". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

[3] Y. Jia, S. Li, X. Guo, B. Lei, J. Hu, X.-H. Xu, and W. Zhang. "Selfee, Self-Supervised Features Extraction of Animal Behaviors". In: *Elife* 11 (2022).

[4] B. Wilson, Y. Chen, D. K. Singh, R. Ojha, J. Pottle, M. Bezick, A. Boltasseva, V. M. Shalaev, and A. V. Kildishev. "Authentication Through Residual Attention-Based Processing of Tampered Optical Responses". In: *Advanced Photonics* 6.5 (2024).

[5] M. Hamilton and T. S. D. Team. *SynapseML: Simple and Distributed Machine Learning*. https://github.com/microsoft/SynapseML. Microsoft, GitHub repository. 2022.

[6] M. Hamilton, N. Gonsalves, C. Lee, A. Raman, B. Walsh, S. Prasad, D. Banda, L. Zhang, L. Zhang, and W. T. Freeman. "Large-Scale Intelligent Microservices". In: *arXiv preprint arXiv:2009.08044* (2020).

[7] M. Hamilton, S. Raghunathan, I. Matiach, A. Schonhoffer, A. Raman, E. Barzilay, K. Rajendran, D. Banda, C. J. Hong, M. Knoertzer, et al. "MMLSpark: Unifying Machine Learning Ecosystems at Massive Scales". In: *arXiv preprint arXiv:1810.08744* (2018).

[8] B. Walsh et al. *Large-Scale Automatic Audiobook Creation*. 2023. arXiv: 2309.03926 [cs.SD]. URL: https://arxiv.org/abs/2309.03926.

[9] M. Hamilton, S. Lundberg, L. Zhang, S. Fu, and W. T. Freeman. "Axiomatic explanations for visual search, retrieval, and similarity learning". In: *arXiv preprint arXiv:2103.00370* (2021).

[10] *The Metropolitan Museum of Art Open Access CSV*. 2019. URL: https://github.com/metmuseum/openaccess.

[11] *The Rijksmuseum Open Access API*. 2019. URL: https://data.rijksmuseum.nl/.

[12] C. E. Thomaz and G. A. Giraldi. "A new ranking method for principal components analysis and its application to face image analysis". In: *Image and vision computing* 28.6 (2010), pp. 902–913.

[13] Y. Matsui, R. Hinami, and S. Satoh. “Reconfigurable Inverted Index”. In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 1715–1723.

[14] V. DeGenova. *Recommending Visually Similar Products Using Content Based Features*. 2017. URL: https://tech.wayfair.com/data-science/2017/12/recommending-visually-similar-products-using-content-based-features/.

[15] Bing. *Beyond text queries: Searching with Bing Visual Search*. 2017. URL: https://blogs.bing.com/search-quality-insights/2017-06/beyond-text-queries-searching-with-bing-visual-search.

[16] C. Mellina. *Introducing Similarity Search at Flickr*. 2017. URL: https://code.flickr.net/2017/03/07/introducing-similarity-search-at-flickr/.

[17] S. Dasgupta and Y. Freund. “Random projection trees and low dimensional manifolds”. In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 537–546.

[18] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

[19] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. “Performance-optimized hierarchical models predict neural responses in higher visual cortex”. In: *Proceedings of the National Academy of Sciences* 111.23 (2014), pp. 8619–8624.

[20] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. “Deep image retrieval: Learning global representations for image search”. In: *European conference on computer vision*. Springer. 2016, pp. 241–257.

[21] Y. Bengio, A. Courville, and P. Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[22] R. Zhang, P. Isola, and A. A. Efros. “Colorful image colorization”. In: *European conference on computer vision*. Springer. 2016, pp. 649–666.

[23] A. Radford, L. Metz, and S. Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[24] G. Koch, R. Zemel, and R. Salakhutdinov. “Siamese neural networks for one-shot image recognition”. In: *ICML deep learning workshop*. Vol. 2. 2015.

[25] M. Huh, P. Agrawal, and A. A. Efros. “What makes ImageNet good for transfer learning?” In: *arXiv preprint arXiv:1608.08614* (2016).

[26] M. Aumüller, E. Bernhardsson, and A. J. Faithfull. “ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms”. In: *CoRR* abs/1807.05614 (2018). arXiv: 1807.05614. URL: http://arxiv.org/abs/1807.05614.

[27] J. L. Bentley. “Multidimensional binary search trees used for associative searching”. In: *Communications of the ACM* 18.9 (1975), pp. 509–517.

[28] Y. Bachrach, Y. Finkelstein, R. Gilad-Bachrach, L. Katzir, N. Koenigstein, N. Nice, and U. Paquet. “Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces”. In: *Proceedings of the 8th ACM Conference on Recommender systems*. 2014, pp. 257–264.

[29] S. M. Omohundro. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

[30] D. Baranchuk, A. Babenko, and Y. Malkov. “Revisiting the Inverted Indices for Billion-Scale Approximate Nearest Neighbors”. In: *CoRR* abs/1802.02422 (2018). arXiv: 1802.02422. URL: http://arxiv.org/abs/1802.02422.

[31] D. Yan, Y. Wang, J. Wang, H. Wang, and Z. Li. “K-nearest Neighbors Search by Random Projection Forests”. In: *IEEE Transactions on Big Data* (2019).

[32] J. Johnson, M. Douze, and H. Jégou. “Billion-scale similarity search with GPUs”. In: *IEEE Transactions on Big Data* (2019).

[33] J. Wang, H. T. Shen, J. Song, and J. Ji. “Hashing for Similarity Search: A Survey”. In: *CoRR* abs/1408.2927 (2014). arXiv: 1408.2927. URL: http://arxiv.org/abs/1408.2927.

[34] K. He, X. Zhang, S. Ren, and J. Sun. “Identity mappings in deep residual networks”. In: *European conference on computer vision*. Springer. 2016, pp. 630–645.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[36] X. Huang and S. Belongie. “Arbitrary style transfer in real-time with adaptive instance normalization”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1501–1510.

[37] C. Olah, A. Mordvintsev, and L. Schubert. “Feature Visualization”. In: *Distill* (2017). https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

[38] E. T. Werner. “Myths and Legends of China. London: George G”. In: *Harrap. Disertasi* (1922).

[39] A. Oppenheim, D. Arnold, D. Arnold, and K. Yamamoto. *Ancient Egypt Transformed: The Middle Kingdom*. Metropolitan Museum of Art, 2015.

[40] W. C. Hayes. *The scepter of Egypt: a background for the study of the Egyptian antiquities in the Metropolitan Museum of Art*. Vol. 1. Metropolitan Museum of Art, 1990.

[41] C. Le Corbeiller. *China Trade Porcelain: Patterns of Exchange: Additions to the Helena Woolworth McCann Collection in the Metropolitan Museum of Art*. Metropolitan Museum of Art, 1974.

[42] T. Volker. *Porcelain and the Dutch East India Company: as recorded in the Dagh-Registers of Batavia Castle, those of Hirado and Deshima and other contemporary papers; 1602-1682*. Vol. 11. Brill Archive, 1954.

[43] A. Fedosejev. *React. js essentials*. Packt Publishing Ltd, 2015.

[44] S. Marcel and Y. Rodriguez. “Torchvision the machine-vision package of torch”. In: *Proceedings of the 18th ACM international conference on Multimedia*. 2010, pp. 1485–1488.

[45] L. A. Gatys, A. S. Ecker, and M. Bethge. “Image style transfer using convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[47] K. Nichol. *Painter by numbers, wikiart*. 2016.

[48] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. “Neural Style Transfer: A Review”. In: *IEEE Transactions on Visualization and Computer Graphics* (2019), pp. 1–1. ISSN: 2160-9306. DOI: 10.1109/tvcg.2019.2921336. URL: http://dx.doi.org/10.1109/tvcg.2019.2921336.

[49] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. “Densenet: Implementing efficient convnet descriptor pyramids”. In: *arXiv preprint arXiv:1404.1869* (2014).

[50] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).

[51] A. Dhesi and P. Kar. “Random projection trees revisited”. In: *Advances in Neural Information Processing Systems*. 2010, pp. 496–504.

[52] D. E. Knuth. *The art of computer programming*. Vol. 3. Pearson Education, 1997.

[53] C. Gormley and Z. Tong. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O’Reilly Media, Inc.", 2015.

[54] M. McCandless, E. Hatcher, O. Gospodnetić, and O. Gospodnetić. *Lucene in action*. Vol. 2. Manning Greenwich, 2010.

[55] J. M. Hellerstein and M. Stonebraker. “Predicate migration: Optimizing queries with expensive predicates”. In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*. 1993, pp. 267–276.

[56] A. Y. Levy, I. S. Mumick, and Y. Sagiv. “Query optimization by predicate move-around”. In: *VLDB*. 1994, pp. 96–107.

[57] S. v. d. Walt, S. C. Colbert, and G. Varoquaux. “The NumPy array: a structure for efficient numerical computation”. In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[59] L. Dagum and R. Menon. “OpenMP: an industry standard API for shared-memory programming”. In: *IEEE computational science and engineering* 5.1 (1998), pp. 46–55.

[60] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. "Cython: The best of both worlds". In: *Computing in Science & Engineering* 13.2 (2011), pp. 31–39.

[61] M. Hamilton, S. Raghunathan, A. Annavajhala, D. Kirsanov, E. Leon, E. Barzilay, I. Matiach, J. Davison, M. Busch, M. Oprescu, et al. "Flexible and Scalable Deep Learning with MMLSpark". In: *International Conference on Predictive Applications and APIs*. 2018, pp. 11–22.

[62] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. "Automatic differentiation in pytorch". In: (2017).

[63] G. Dinu, A. Lazaridou, and M. Baroni. "Improving zero-shot learning by mitigating the hubness problem". In: *arXiv preprint arXiv:1412.6568* (2014).

[64] X. Wang. "A fast exact k-nearest neighbors algorithm for high dimensional search using k-means clustering and triangle inequality". In: *The 2011 International Joint Conference on Neural Networks*. IEEE. 2011, pp. 1293–1299.

[65] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6626–6637.

[66] T. Karras, T. Aila, S. Laine, and J. Lehtinen. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. 2017. arXiv: 1710.10196 [cs.NE].

[67] Z. Liu, P. Luo, X. Wang, and X. Tang. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015.

[68] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, H. Strobelt, B. Zhou, and A. Torralba. *Seeing What a GAN Cannot Generate*. 2019. arXiv: 1910.11626 [cs.CV].

[69] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation". In: *CoRR* abs/1801.04381 (2018). arXiv: 1801.04381. URL: http://arxiv.org/abs/1801.04381.

[70] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. "Aggregated Residual Transformations for Deep Neural Networks". In: *CoRR* abs/1611.05431 (2016). arXiv: 1611.05431. URL: http://arxiv.org/abs/1611.05431.

[71] L. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking Atrous Convolution for Semantic Image Segmentation". In: *CoRR* abs/1706.05587 (2017). arXiv: 1706.05587. URL: http://arxiv.org/abs/1706.05587.

[72] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870.

[73] N. Bhatia et al. "Survey of nearest neighbor techniques". In: *arXiv preprint arXiv:1007.0085* (2010).

[74] E. Marchiori. "Class conditional nearest neighbor for large margin instance selection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.2 (2009), pp. 364–370.

[75] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[76] A. Veit, S. Belongie, and T. Karaletsos. *Conditional Similarity Networks*. 2016. arXiv: 1603.07810 [cs.CV].

[77] P. Lu, G. Huang, Y. Fu, G. Guo, and H. Lin. "Learning Large Euclidean Margin for Sketch-based Image Retrieval". In: *CoRR* abs/1812.04275 (2018). arXiv: 1812.04275. URL: http://arxiv.org/abs/1812.04275.

[78] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song. "Neural Style Transfer: A Review". In: *CoRR* abs/1705.04058 (2017). arXiv: 1705.04058. URL: http://arxiv.org/abs/1705.04058.

[79] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. "Visual attribute transfer through deep image analogy". In: *arXiv preprint arXiv:1705.01088* (2017).

[80] C. Traina, A. J. M. Trains, and J. M. de Figuciredo. "Including conditional operators in content-based image retrieval in large sets of medical exams". In: *Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems*. 2004, pp. 85–90.

[81] B. A. Plummer, P. Kordas, M. Hadi Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik. "Conditional image-text embedding networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 249–264.

[82] X. Gao, T. Mu, J. Y. Goulermas, J. Thiyagalingam, and M. Wang. "An Interpretable Deep Architecture for Similarity Learning Built Upon Hierarchical Concepts". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 3911–3926.

[83] L. Liao, X. He, B. Zhao, C.-W. Ngo, and T.-S. Chua. "Interpretable multimodal retrieval for fashion products". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 1571–1579.

[84] H. Caesar, J. Uijlings, and V. Ferrari. "Coco-stuff: Thing and stuff classes in context". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1209–1218.

[85] J. H. Cho, U. Mall, K. Bala, and B. Hariharan. "PiCIE: Unsupervised Semantic Segmentation using Invariance and Equivariance in Clustering". In: *ArXiv* abs/2103.17070 (2021).

[86] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9650–9660.

[87] A. Zlateski, R. Jaroensri, P. Sharma, and F. Durand. "On the importance of label quality for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1479–1487.

[88] H. Yu, Z. Yang, L. Tan, Y. Wang, W. Sun, M. Sun, and Y. Tang. "Methods and datasets on semantic segmentation: A review". In: *Neurocomputing* 304 (2018), pp. 82–103.

[89] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, A. G. Schwing, and J. Kautz. “UFO2: A Unified Framework Towards Omni-supervised Object Detection”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 288–313.

[90] S.-Y. Pan, C.-Y. Lu, S.-P. Lee, and W.-H. Peng. “Weakly-Supervised Image Semantic Segmentation Using Graph Convolutional Networks”. In: *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2021, pp. 1–6.

[91] Y. Liu, Y.-H. Wu, P. Wen, Y. Shi, Y. Qiu, and M.-M. Cheng. “Leveraging Instance-, Image- and Dataset-Level Information for Weakly Supervised Instance Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020). DOI: 10.1109/TPAMI.2020.3023152.

[92] H. Bilen, R. Benenson, and S. J. Oh. *ECCV 2020 Tutorial on Weakly-Supervised Learning in Computer Vision*. URL: https://github.com/hbilen/wsl-eccv20.github.io.

[93] X. Ji, J. F. Henriques, and A. Vedaldi. “Invariant information clustering for unsupervised image classification and segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9865–9874.

[94] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1096–1103.

[95] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. “Context Encoders: Feature Learning by Inpainting”. In: *CVPR*. 2016.

[96] R. Zhang, P. Isola, and A. A. Efros. “Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction”. In: *CVPR*. 2017.

[97] S. Gidaris, P. Singh, and N. Komodakis. “Unsupervised representation learning by predicting image rotations”. In: *arXiv preprint arXiv:1803.07728* (2018).

[98] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. “Learning deep representations by mutual information estimation and maximization”. In: *arXiv preprint arXiv:1808.06670* (2018).

[99] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. “Big Self-Supervised Models are Strong Semi-Supervised Learners”. In: *arXiv preprint arXiv:2006.10029* (2020).

[100] X. Chen, H. Fan, R. Girshick, and K. He. “Improved Baselines with Momentum Contrastive Learning”. In: *arXiv preprint arXiv:2003.04297* (2020).

[101] A. v. d. Oord, Y. Li, and O. Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

[102] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. “Unsupervised feature learning via non-parametric instance discrimination”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3733–3742.

[103] P. O. Pinheiro, A. Almahairi, R. Y. Benmalek, F. Golemo, and A. Courville. “Unsupervised learning of dense visual representations”. In: *arXiv preprint arXiv:2011.05499* (2020).

[104] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng. “Contrastive Clustering”. In: *arXiv preprint arXiv:2009.09687* (2020).

[105] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. “Scan: Learning to classify images without labels”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 268–285.

[106] J. Hwang, S. X. Yu, J. Shi, M. D. Collins, T. Yang, X. Zhang, and L. Chen. “SegSort: Segmentation by Discriminative Sorting of Segments”. In: *CoRR* abs/1910.06962 (2019). arXiv: 1910.06962. URL: http://arxiv.org/abs/1910.06962.

[107] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. Van Gool. “Unsupervised Semantic Segmentation by Contrasting Object Mask Proposals”. In: *arxiv preprint arxiv:2102.06191* (2021).

[108] E. Collins, R. Achanta, and S. Susstrunk. “Deep feature factorization for concept discovery”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 336–352.

[109] X. Wang, R. Girshick, A. Gupta, and K. He. “Non-local neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803.

[110] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. “Self-attention generative adversarial networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 7354–7363.

[111] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[112] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[113] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. “Training data-efficient image transformers & distillation through attention”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10347–10357.

[114] R. Ranftl, A. Bochkovskiy, and V. Koltun. “Vision Transformers for Dense Prediction”. In: *ArXiv preprint* (2021).

[115] M. Hamilton, S. Lundberg, L. Zhang, S. Fu, and W. T. Freeman. “Model-Agnostic Explainability for Visual Search”. In: *arXiv preprint arXiv:2103.00370* (2021).

[116] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[117] Z. Teed and J. Deng. “Raft: Recurrent all-pairs field transforms for optical flow”. In: *European conference on computer vision*. Springer. 2020, pp. 402–419.

[118] P. Krähenbühl and V. Koltun. "Efficient inference in fully connected crfs with gaussian edge potentials". In: *Advances in neural information processing systems* 24 (2011), pp. 109–117.

[119] X. Glorot, A. Bordes, and Y. Bengio. "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.

[120] J. MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[121] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[122] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. "Deep clustering for unsupervised learning of visual features". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 132–149.

[123] G. LoweDavid. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* (2004).

[124] C. Doersch, A. Gupta, and A. A. Efros. "Unsupervised visual representation learning by context prediction". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1422–1430.

[125] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. "Learning visual groups from co-occurrences in space and time". In: *arXiv preprint arXiv:1511.06811* (2015).

[126] Y. Ouali, C. Hudelot, and M. Tami. "Autoregressive unsupervised image segmentation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 142–158.

[127] S. E. Mirsadeghi, A. Royat, and H. Rezatofighi. "Unsupervised Image Segmentation by Mutual Information Maximization and Adversarial Regularization". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6931–6938.

[128] R. B. Potts. "Some generalized order-disorder transformations". In: *Mathematical proceedings of the cambridge philosophical society*. Vol. 48. Cambridge University Press. 1952, pp. 106–109.

[129] G. A. Baker Jr and J. M. Kincaid. "Continuous-Spin Ising Model and λ: ϕ 4: d Field Theory". In: *Physical Review Letters* 42.22 (1979), p. 1431.

[130] G. E. Hinton. "Training products of experts by minimizing contrastive divergence". In: *Neural computation* 14.8 (2002), pp. 1771–1800.

[131] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035.

[132] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: http://tensorflow.org/.

[133] X. Chen, H. Fan, R. Girshick, and K. He. "Improved baselines with momentum contrastive learning". In: *arXiv preprint arXiv:2003.04297* (2020).

[134] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[135] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[136] K. Weiss, T. M. Khoshgoftaar, and D. Wang. "A survey of transfer learning". In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[137] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. *Momentum Contrast for Unsupervised Visual Representation Learning*. 2019. DOI: 10.48550/ARXIV.1911.05722. URL: https://arxiv.org/abs/1911.05722.

[138] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[139] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[140] A. Radford and K. Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018.

[141] S. Schneider, A. Baevski, R. Collobert, and M. Auli. *wav2vec: Unsupervised Pre-training for Speech Recognition*. 2019. DOI: 10.48550/ARXIV.1904.05862. URL: https://arxiv.org/abs/1904.05862.

[142] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. 2021. DOI: 10.48550/ARXIV.2106.07447. URL: https://arxiv.org/abs/2106.07447.

[143] L. Shao, F. Zhu, and X. Li. "Transfer learning for visual categorization: A survey". In: *IEEE transactions on neural networks and learning systems* 26.5 (2014), pp. 1019–1034.

[144] J. Ahn, S. Cho, and S. Kwak. "Weakly supervised learning of instance segmentation with inter-pixel relations". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2209–2218.

[145] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman. "Unsupervised Semantic Segmentation by Distilling Feature Correspondences". In: *arXiv preprint arXiv:2203.08414* (2022).

[146] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen. "Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12275–12284.

[147] C. Liu, J. Yuen, and A. Torralba. "Sift flow: Dense correspondence across scenes and its applications". In: *IEEE transactions on pattern analysis and machine intelligence* 33.5 (2010), pp. 978–994.

[148] S. Kobayashi, E. Matsumoto, and V. Sitzmann. "Decomposing NeRF for Editing via Feature Field Distillation". In: *arXiv preprint arXiv:2205.15585* (2022).

[149] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. DOI: 10.48550/ARXIV.2112.10752. URL: https://arxiv.org/abs/2112.10752.

[150] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[151] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[152] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 2020. DOI: 10.48550/ARXIV.2003.08934. URL: https://arxiv.org/abs/2003.08934.

[153] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. "Joint Bilateral Upsampling". In: *ACM Trans. Graph.* 26.3 (2007), 96–es. ISSN: 0730-0301. DOI: 10.1145/1276377.1276497. URL: https://doi.org/10.1145/1276377.1276497.

[154] J. R. Lee, S. Kim, I. Park, T. Eo, and D. Hwang. "Relevance-cam: Your model already knows where to look". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14944–14953.

[155] Z. Qin, D. Kim, and T. Gedeon. "Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator". In: *arXiv preprint arXiv:1911.10688* (2019).

[156] C. Tomasi and R. Manduchi. "Bilateral filtering for gray and color images". In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815.

[157] L. Caraffa, J.-P. Tarel, and P. Charbonnier. "The Guided Bilateral Filter: When the Joint/Cross Bilateral Filter Becomes Robust". In: *IEEE Transactions on Image Processing* 24.4 (2015), pp. 1199–1208. DOI: 10.1109/TIP.2015.2389617.

[158] C. Xiao and J. Gan. "Fast Image Dehazing Using Guided Joint Bilateral Filter". In: *Vis. Comput.* 28.6–8 (2012), pp. 713–721. ISSN: 0178-2789. DOI: 10.1007/s00371-012-0679-y. URL: https://doi.org/10.1007/s00371-012-0679-y.

[159] D. Mazzini. *Guided Upsampling Network for Real-Time Semantic Segmentation*. 2018. DOI: 10.48550/ARXIV.1807.07466. URL: https://arxiv.org/abs/1807.07466.

[160] A. Buades, B. Coll, and J.-M. Morel. "A non-local algorithm for image denoising". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. 2005, 60–65 vol. 2. DOI: 10.1109/CVPR.2005.38.

[161] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler. *Superpixel Convolutional Networks using Bilateral Inceptions*. 2015. DOI: 10.48550/ARXIV.1511.06739. URL: https://arxiv.org/abs/1511.06739.

[162] X. Wang, R. Girshick, A. Gupta, and K. He. *Non-local Neural Networks*. 2017. DOI: 10.48550/ARXIV.1711.07971. URL: https://arxiv.org/abs/1711.07971.

[163] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand. "Deep bilateral learning for real-time image enhancement". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), p. 118.

[164] T.-M. Li, M. Gharbi, A. Adams, F. Durand, and J. Ragan-Kelley. "Differentiable programming for image processing and deep learning in Halide". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 37.4 (2018), 139:1–139:13.

[165] S. Qian, H. Shao, Y. Zhu, M. Li, and J. Jia. "Blending Anti-Aliasing into Vision Transformer". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 5416–5429. URL: https://proceedings.neurips.cc/paper/2021/file/2b3bf3eee2475e03885a110e9acaab61-Paper.pdf.

[166] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng. "Transformer for Single Image Super-Resolution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2022, pp. 457–466.

[167] W. Freeman and A. Torralba. "Shape recipes: Scene representations that refer to the image". In: *Advances in Neural Information Processing Systems* 15 (2002).

[168] H. Su, V. Jampani, D. Sun, O. Gallo, E. G. Learned-Miller, and J. Kautz. "Pixel-Adaptive Convolutional Neural Networks". In: *CoRR* abs/1904.05373 (2019). arXiv: 1904.05373. URL: http://arxiv.org/abs/1904.05373.

[169] N. Araslanov and S. Roth. "Single-Stage Semantic Segmentation From Image Labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[170] T. Prangemeier, C. Reich, and H. Koeppl. "Attention-Based Transformers for Instance Segmentation of Cells in Microstructures". In: *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2020, pp. 700–707. DOI: 10.1109/BIBM49941.2020.9313305.

[171] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon. *Semantically-Guided Representation Learning for Self-Supervised Monocular Depth*. 2020. DOI: 10.48550/ARXIV.2002.12319. URL: https://arxiv.org/abs/2002.12319.

[172] J. Choi, D. Jung, Y. Lee, D. Kim, D. Manocha, and D. Lee. "SelfDeco: Self-Supervised Monocular Depth Completion in Challenging Indoor Environments". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 467–474. DOI: 10.1109/ICRA48506.2021.9560831.

[173] H. Choi, H. Lee, S. Kim, S. Kim, S. Kim, K. Sohn, and D. Min. *Adaptive confidence thresholding for monocular depth estimation*. 2020. DOI: 10.48550/ARXIV.2009.12840. URL: https://arxiv.org/abs/2009.12840.

[174] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka. "SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation". In: *Computer Vision – ECCV 2020*. Ed. by A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm. Cham: Springer International Publishing, 2020, pp. 1–19. ISBN: 978-3-030-58604-1.

[175] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler. "Superpixel convolutional networks using bilateral inceptions". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 597–613.

[176] A. Shocher, N. Cohen, and M. Irani. "Zero-Shot Super-Resolution Using Deep Internal Learning". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3118–3126. DOI: 10.1109/CVPR.2018.00329.

[177] Y. Chen, S. Liu, and X. Wang. "Learning continuous image representation with local implicit image function". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8628–8638.

[178] D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Deep Image Prior". In: *International Journal of Computer Vision* 128.7 (2020), pp. 1867–1888. DOI: 10.1007/s11263-020-01303-4. URL: https://doi.org/10.1007%2Fs11263-020-01303-4.

[179] R. Keys. "Cubic convolution interpolation for digital image processing". In: *IEEE transactions on acoustics, speech, and signal processing* 29.6 (1981), pp. 1153–1160.

[180] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[181] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel. *Deep ViT Features as Dense Visual Descriptors*. 2021. DOI: 10.48550/ARXIV.2112.05814. URL: https://arxiv.org/abs/2112.05814.

[182] N. Tumanyan, O. Bar-Tal, S. Bagon, and T. Dekel. *Splicing ViT Features for Semantic Appearance Transfer*. 2022. DOI: 10.1109/CVPR52688.2022.01048.

[183] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang. *Is the deconvolution layer the same as a convolutional layer?* 2016. DOI: 10.48550/ARXIV.1609.07009. URL: https://arxiv.org/abs/1609.07009.

[184] V. Dumoulin and F. Visin. *A guide to convolution arithmetic for deep learning*. 2016. DOI: 10.48550/ARXIV.1603.07285. URL: https://arxiv.org/abs/1603.07285.

[185] H. Noh, S. Hong, and B. Han. "Learning Deconvolution Network for Semantic Segmentation". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 1520–1528.

[186] J. Johnson, A. Alahi, and L. Fei-Fei. "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Cham: Springer International Publishing, 2016, pp. 694–711.

[187] V. Dumoulin and F. Visin. "A guide to convolution arithmetic for deep learning". In: *arXiv preprint arXiv:1603.07285* (2016).

[188] A. Odena, V. Dumoulin, and C. Olah. “Deconvolution and Checkerboard Artifacts”. In: *Distill* (2016). DOI: 10.23915/distill.00003. URL: http://distill.pub/2016/deconv-checkerboard.

[189] J. Gauthier. “Conditional generative adversarial nets for convolutional face generation”. In: 2015.

[190] C. Dong, C. C. Loy, K. He, and X. Tang. *Image Super-Resolution Using Deep Convolutional Networks*. 2015. DOI: 10.48550/ARXIV.1501.00092. URL: https://arxiv.org/abs/1501.00092.

[191] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[192] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng. “H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes”. In: *IEEE Transactions on Medical Imaging* 37.12 (2018), pp. 2663–2674. DOI: 10.1109/TMI.2018.2845918.

[193] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu. “UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 1055–1059. DOI: 10.1109/ICASSP40776.2020.9053405.

[194] J. Fu, J. Liu, Y. Li, Y. Bao, W. Yan, Z. Fang, and H. Lu. “Contextual deconvolution network for semantic segmentation”. In: *Pattern Recognition* 101 (2020), p. 107152. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2019.107152. URL: https://www.sciencedirect.com/science/article/pii/S0031320319304534.

[195] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. *Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution*. 2017. DOI: 10.48550/ARXIV.1704.03915. URL: https://arxiv.org/abs/1704.03915.

[196] T. Tong, G. Li, X. Liu, and Q. Gao. “Image super-resolution using dense skip connections”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4799–4807.

[197] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[198] H. Lu, Y. Dai, C. Shen, and S. Xu. “Index Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.1 (2022), pp. 242–255. DOI: 10.1109/TPAMI.2020.3004474.

[199] Y. Dai, H. Lu, and C. Shen. *Learning Affinity-Aware Upsampling for Deep Image Matting*. 2020. arXiv: 2011.14288 `[cs.CV]`.

[200] H. Lu, W. Liu, H. Fu, and Z. Cao. “FADE: Fusing the Assets of Decoder and Encoder for Task-Agnostic Upsampling”. In: *Proc. European Conference on Computer Vision (ECCV)*. 2022.

[201] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin. "CARAFE: Content-Aware ReAssembly of FEatures". In: (2019). DOI: 10.48550/ARXIV.1905.02188. URL: https://arxiv.org/abs/1905.02188.

[202] H. Lu, W. Liu, Z. Ye, H. Fu, Y. Liu, and Z. Cao. "SAPA: Similarity-Aware Point Affiliation for Feature Upsampling". In: *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2022.

[203] H. Wu, S. Zheng, J. Zhang, and K. Huang. *Fast End-to-End Trainable Guided Filter*. 2019. arXiv: 1803.05619 [cs.CV].

[204] H. Hu, Y. Chen, J. Xu, S. Borse, H. Cai, F. Porikli, and X. Wang. *Learning Implicit Feature Alignment Function for Semantic Segmentation*. 2022. arXiv: 2206.08655 [cs.CV].

[205] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. "Implicit Neural Representations with Periodic Activation Functions". In: *Proc. NeurIPS*. 2020.

[206] Z. Chen and H. Zhang. "Learning implicit fields for generative shape modeling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5939–5948.

[207] M. Hamilton, E. Shelhamer, and W. T. Freeman. "It Is Likely That Your Loss Should be a Likelihood". In: *arXiv preprint arXiv:2007.06059* (2020).

[208] D. Hendrycks and K. Gimpel. "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415* (2016).

[209] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. *Implicit Neural Representations with Periodic Activation Functions*. 2020. DOI: 10.48550/ARXIV.2006.09661. URL: https://arxiv.org/abs/2006.09661.

[210] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*. 2020. DOI: 10.48550/ARXIV.2006.10739. URL: https://arxiv.org/abs/2006.10739.

[211] X. Glorot, A. Bordes, and Y. Bengio. "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.

[212] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[213] J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[214] W. B. Johnson, J. Lindenstrauss, and G. Schechtman. "Extensions of Lipschitz maps into Banach spaces". In: *Israel Journal of Mathematics* 54.2 (1986), pp. 129–138.

[215] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: nonlinear phenomena* 60.1-4 (1992), pp. 259–268.

[216] G. Alain and Y. Bengio. *Understanding intermediate layers using linear classifier probes*. 2016. DOI: 10.48550/ARXIV.1610.01644. URL: https://arxiv.org/abs/1610.01644.

[217] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer”. In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[218] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. “SegFormer: Simple and efficient design for semantic segmentation with transformers”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.

[219] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. “Semantic understanding of scenes through the ade20k dataset”. In: *International Journal of Computer Vision* 127.3 (2019), pp. 302–321.

[220] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. “Scene Parsing through ADE20K Dataset”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[221] W. Liu, H. Lu, H. Fu, and Z. Cao. *Learning to Upsample by Learning to Sample*. 2023. arXiv: 2308.15085 [cs.CV].

[222] Y. Dai, H. Lu, and C. Shen. “Learning affinity-aware upsampling for deep image matting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6841–6850.

[223] L. Smith and C. Yu. “Infants rapidly learn word-referent mappings via cross-situational statistics”. In: *Cognition* 106.3 (2008), pp. 1558–1568.

[224] N. Chomsky. *Language and problems of knowledge: The Managua lectures*. Vol. 16. MIT press, 1987.

[225] G. K. Pullum and B. C. Scholz. “Empirical assessment of stimulus poverty arguments”. In: *The linguistic review* 19.1-2 (2002), pp. 9–50.

[226] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. “Emerging Properties in Self-Supervised Vision Transformers”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2021.

[227] Y.-J. Shih, H.-F. Wang, H.-J. Chang, L. Berry, H.-y. Lee, and D. Harwath. “Speechclip: Integrating speech with pre-trained vision and language model”. In: *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2023, pp. 715–722.

[228] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[229] Y. Gong, A. Rouditchenko, A. H. Liu, D. Harwath, L. Karlinsky, H. Kuehne, and J. R. Glass. “Contrastive audio-visual masked autoencoder”. In: *The Eleventh International Conference on Learning Representations*. 2022.

[230] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. “Scene parsing through ade20k dataset”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 633–641.

[231] D. Harwath, A. Recasens, D. Surís, G. Chuang, A. Torralba, and J. Glass. “Jointly discovering visual objects and spoken words from raw sensory input”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 649–665.

[232] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. “Is object localization for free?-weakly-supervised learning with convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 685–694.

[233] R. Arandjelovic and A. Zisserman. “Objects that sound”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 435–451.

[234] R. C. Fong and A. Vedaldi. “Interpretable explanations of black boxes by meaningful perturbation”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 3429–3437.

[235] G. A. Miller. “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[236] Z. Wu and M. Palmer. “Verb semantics and lexical selection”. In: *arXiv preprint cmp-lg/9406033* (1994).

[237] J. W. Fisher III, T. Darrell, W. Freeman, and P. Viola. “Learning joint statistical models for audio-visual fusion and segregation”. In: *Advances in neural information processing systems* 13 (2000).

[238] J. W. Fisher and T. Darrell. “Probabalistic models and informative subspaces for audiovisual correspondence”. In: *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part III 7*. Springer. 2002, pp. 592–603.

[239] H. Zhu, M.-D. Luo, R. Wang, A.-H. Zheng, and R. He. “Deep audio-visual learning: A survey”. In: *International Journal of Automation and Computing* 18 (2021), pp. 351–376.

[240] S. Chopra, R. Hadsell, and Y. LeCun. “Learning a similarity metric discriminatively, with application to face verification”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. IEEE. 2005, pp. 539–546.

[241] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. “A survey on contrastive self-supervised learning”. In: *Technologies* 9.1 (2020), p. 2.

[242] H. Chen, W. Xie, T. Afouras, A. Nagrani, A. Vedaldi, and A. Zisserman. “Localizing visual sounds the hard way”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16867–16876.

[243] P. Peng and D. Harwath. “Self-supervised representation learning for speech using visual grounding and masked language modeling”. In: *arXiv preprint arXiv:2202.03543* (2022).

[244] P. Peng and D. Harwath. “Word discovery in visually grounded, self-supervised speech models”. In: *arXiv preprint arXiv:2203.15081* (2022).

[245] L. Wang and A. v. d. Oord. “Multi-format contrastive learning of audio representations”. In: *arXiv preprint arXiv:2103.06508* (2021).

[246] M. Monfort, S. Jin, A. Liu, D. Harwath, R. Feris, J. Glass, and A. Oliva. "Spoken moments: Learning joint audio-visual representations from video descriptions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14871–14881.

[247] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. "Imagebind: One embedding space to bind them all". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 15180–15190.

[248] S. Ma, Z. Zeng, D. McDuff, and Y. Song. "Active contrastive learning of audio-visual video representations". In: *arXiv preprint arXiv:2009.09805* (2020).

[249] A. S. Park and J. R. Glass. "Unsupervised pattern discovery in speech". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.1 (2007), pp. 186–197.

[250] X. Ji, J. F. Henriques, and A. Vedaldi. "Invariant information clustering for unsupervised image classification and segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9865–9874.

[251] J. H. Cho, U. Mall, K. Bala, and B. Hariharan. "Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16794–16804.

[252] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. "Deep Clustering for Unsupervised Learning of Visual Features". In: *European Conference on Computer Vision*. 2018.

[253] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman. "Unsupervised semantic segmentation by distilling feature correspondences". In: *arXiv preprint arXiv:2203.08414* (2022).

[254] D. Harwath and J. R. Glass. "Learning word-like units from joint audio-visual analysis". In: *arXiv preprint arXiv:1701.07481* (2017).

[255] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran. "Self-supervised learning by cross-modal audio-video clustering". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9758–9770.

[256] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. "The sound of pixels". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 570–586.

[257] N. Kumar, S. Goel, A. Narang, and M. Hasan. "Robust one shot audio to video generation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 770–771.

[258] S. Liu, S. Li, and H. Cheng. "Towards an end-to-end visual-to-raw-audio generation with GAN". In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.3 (2021), pp. 1299–1312.

[259] J. Choi, J. Hong, and Y. M. Ro. "DiffV2S: Diffusion-based Video-to-Speech Synthesis with Vision-guided Speaker Embedding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 7812–7821.

[260] Y. Mao, J. Zhang, M. Xiang, Y. Lv, Y. Zhong, and Y. Dai. "Contrastive conditional latent diffusion for audio-visual segmentation". In: *arXiv preprint arXiv:2307.16579* (2023).

[261] Y. Tian, D. Krishnan, and P. Isola. "Contrastive multiview coding". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer. 2020, pp. 776–794.

[262] A. Guzhov, F. Raue, J. Hees, and A. Dengel. "Audioclip: Extending clip to image, text and audio". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 976–980.

[263] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[264] M. Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2023.

[265] B. Psomas, I. Kakogeorgiou, K. Karantzalos, and Y. Avrithis. "Keep It SimPool: Who Said Supervised Transformers Suffer from Attention Deficit?" In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 5350–5360.

[266] N. Frosst, N. Papernot, and G. Hinton. "Analyzing and improving representations with the soft nearest neighbor loss". In: *International conference on machine learning*. PMLR. 2019, pp. 2012–2020.

[267] T. Wang and P. Isola. "Understanding contrastive representation learning through alignment and uniformity on the hypersphere". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9929–9939.

[268] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. "Vision Transformers Need Registers". In: *arXiv preprint arXiv:2309.16588* (2023).

[269] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. "Hubert: Self-supervised speech representation learning by masked prediction of hidden units". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), pp. 3451–3460.

[270] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. "Librispeech: an asr corpus based on public domain audio books". In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2015, pp. 5206–5210.

[271] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems* 27 (2014).

[272] A. E. Hoerl and R. W. Kennard. "Ridge regression: Biased estimation for nonorthogonal problems". In: *Technometrics* 12.1 (1970), pp. 55–67.

[273] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. "Audio set: An ontology and human-labeled dataset for audio events". In: *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2017, pp. 776–780.

[274] D. Harwath, A. Torralba, and J. Glass. “Unsupervised learning of spoken language with visual context”. In: *Advances in Neural Information Processing Systems* 29 (2016).

[275] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685* (2021).

[276] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. “Fastspeech 2: Fast and high-quality end-to-end text to speech”. In: *arXiv preprint arXiv:2006.04558* (2020).

[277] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman. “Vggsound: A large-scale audio-visual dataset”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 721–725.

[278] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 `[cs.CL]`.

[279] H. W. Kuhn. “The Hungarian method for the assignment problem”. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[280] P. Nayak. *Understanding searches better than ever before*. 2019. URL: https://blog.google/products/search/search-language-understanding-bert/.

[281] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *arXiv preprint arXiv:2002.05709* (2020).

[282] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. “Learning transferable visual models from natural language supervision”. In: *arXiv preprint arXiv:2103.00020* (2021).

[283] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. “Unsupervised learning of visual features by contrasting cluster assignments”. In: *arXiv preprint arXiv:2006.09882* (2020).

[284] A. Mowshowitz and A. Kawaguchi. “Assessing bias in search engines”. In: *Information Processing & Management* 38.1 (2002), pp. 141–156.

[285] A. Diaz. “Through the Google goggles: Sociopolitical bias in search engine design”. In: *Web search*. Springer, 2008, pp. 11–34.

[286] E. Goldman. “Search engine bias and the demise of search engine utopianism”. In: *Yale JL & Tech.* 8 (2005), p. 188.

[287] S. Zhu, T. Yang, and C. Chen. “Visual explanation for deep metric learning”. In: *arXiv preprint arXiv:1909.12977* (2019).

[288] M. Zheng, S. Karanam, T. Chen, R. J. Radke, and Z. Wu. “Towards Visually Explaining Similarity Models”. In: *arXiv preprint arXiv:2008.06035* (2020).

[289] B. Dong, R. Collins, and A. Hoogs. “Explainability for Content-Based Image Retrieval.” In.

[290] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[291] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. “Attention is all you need”. In: *arXiv preprint arXiv:1706.03762* (2017).

[292] M. Sundararajan, K. Dhamdhere, and A. Agarwal. “The Shapley Taylor Interaction Index”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9259–9268.

[293] J. C. Harsanyi. “A simplified bargaining model for the n-person cooperative game”. In: *International Economic Review* 4.2 (1963), pp. 194–220.

[294] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. “Learning Deep Features for Discriminative Localization.” In: *CVPR* (2016).

[295] M. T. Ribeiro, S. Singh, and C. Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, pp. 1135–1144.

[296] S. M. Lundberg and S.-I. Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 4765–4774.

[297] X. Su and T. M. Khoshgoftaar. “A survey of collaborative filtering techniques”. In: *Advances in artificial intelligence* 2009 (2009).

[298] R. Kohavi, D. H. Wolpert, et al. “Bias plus variance decomposition for zero-one loss functions”. In: *ICML*. Vol. 96. 1996, pp. 275–83.

[299] Y. Koren. “The bellkor solution to the netflix grand prize”. In: *Netflix prize documentation* 81.2009 (2009), pp. 1–10.

[300] H. Nori, S. Jenkins, P. Koch, and R. Caruana. “Interpretml: A unified framework for machine learning interpretability”. In: *arXiv preprint arXiv:1909.09223* (2019).

[301] T. J. Hastie and R. J. Tibshirani. *Generalized additive models*. Vol. 43. CRC press, 1990.

[302] L. S. Shapley. “Notes on the n-Person Game—II: The Value of an n-Person Game”. In: (1951).

[303] A. Shrikumar, P. Greenside, and A. Kundaje. *Learning Important Features Through Propagating Activation Differences*. 2017. arXiv: 1704.02685 [cs.CV].

[304] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation”. In: *PloS one* 10.7 (2015), e0130140.

[305] E. Štrumbelj and I. Kononenko. "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and information systems* 41.3 (2014), pp. 647–665.

[306] A. Datta, S. Sen, and Y. Zick. "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems". In: *2016 IEEE symposium on security and privacy (SP)*. IEEE. 2016, pp. 598–617.

[307] S. Lipovetsky and M. Conklin. "Analysis of regression in game theory approach". In: *Applied Stochastic Models in Business and Industry* 17.4 (2001), pp. 319–330.

[308] A. Saabas. *Interpreting random forests*. 2014. URL: http://blog.datadive.net/interpreting-random-forests/.

[309] C. Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[310] H. P. Young. "Monotonic solutions of cooperative games". In: *International Journal of Game Theory* 14.2 (1985), pp. 65–72.

[311] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. "Counterfactual visual explanations". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2376–2384.

[312] H. Chefer, S. Gur, and L. Wolf. "Transformer interpretability beyond attention visualization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 782–791.

[313] J. Singh and A. Anand. "Exs: Explainable search using local model agnostic interpretability". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 2019, pp. 770–773.

[314] Z. T. Fernando, J. Singh, and A. Anand. "A study on the Interpretability of Neural Retrieval Models using DeepSHAP". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 1005–1008.

[315] M. Ancona, C. Oztireli, and M. Gross. "Explaining deep neural networks with a polynomial time algorithm for shapley value approximation". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 272–281.

[316] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Van der Smagt, D. Cremers, and T. Brox. "Flownet: Learning optical flow with convolutional networks". In: *arXiv preprint arXiv:1504.06852* (2015).

[317] G. Sun, W. Wang, J. Dai, and L. Van Gool. "Mining cross-image semantics for weakly supervised semantic segmentation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 347–365.

[318] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely. "Learning Feature Descriptors using Camera Pose Supervision". In: *Proc. European Conference on Computer Vision (ECCV)*. 2020.

[319] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang. "Show, Match and Segment: Joint Weakly Supervised Learning of Semantic Matching and Object Co-segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (2020).

[320] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen. “Cross attention network for few-shot classification”. In: *arXiv preprint arXiv:1910.07677* (2019).

[321] X. Wei, T. Zhang, Y. Li, Y. Zhang, and F. Wu. “Multi-modality cross attention network for image and sentence matching”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10941–10950.

[322] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li. “Axiom-based grad-cam: Towards accurate visualization and explanation of cnns”. In: *arXiv preprint arXiv:2008.02312* (2020).

[323] G. Owen. “Multilinear extensions of games”. In: *Management Science* 18.5-part-2 (1972), pp. 64–79.

[324] J. D. Janizek, P. Sturmfels, and S.-I. Lee. “Explaining explanations: Axiomatic feature interactions for deep networks”. In: *arXiv preprint arXiv:2002.04138* (2020).

[325] S. Lundberg. *shap*. URL: https://github.com/slundberg/shap.

[326] M. Sundararajan, A. Taly, and Q. Yan. “Axiomatic attribution for deep networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3319–3328.

[327] R. J. Aumann and L. S. Shapley. *Values of non-atomic games*. Princeton University Press, 2015.

[328] R. Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[329] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Van der Smagt, D. Cremers, and T. Brox. “Flownet: Learning optical flow with convolutional networks”. In: *arXiv preprint arXiv:1504.06852* (2015).

[330] G. Paolacci, J. Chandler, and P. G. Ipeirotis. “Running experiments on amazon mechanical turk”. In: *Judgment and Decision making* 5.5 (2010), pp. 411–419.

[331] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[332] M. Hamilton, S. Fu, W. T. Freeman, and M. Lu. “Conditional Image Retrieval”. In: *arXiv preprint arXiv:2007.07177* (2020).

[333] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[334] K. Simonyan and A. Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[335] M. Sundararajan, A. Taly, and Q. Yan. “Axiomatic attribution for deep networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3319–3328.

[336] V. Sobal, M. Ibrahim, R. Balestriero, V. Cabannes, D. Bouchacourt, P. Astolfi, K. Cho, and Y. LeCun. “X-Sample Contrastive Loss: Improving Contrastive Learning with Sample Similarity Graphs”. In: *International Conference on Learning Representations* (2025).

[337] T. Hu, Z. Liu, F. Zhou, W. Wang, and W. Huang. "Your Contrastive Learning Is Secretly Doing Stochastic Neighbor Embedding". In: *International Conference on Learning Representations*. 2023.

[338] J. Yang, C. Li, P. Zhang, B. Xiao, C. Liu, L. Yuan, and J. Gao. "Unified contrastive learning in image-text-label space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19163–19173.

[339] J. N. Böhm, P. Berens, and D. Kobak. "Unsupervised visualization of image datasets using contrastive learning". In: *International Conference on Learning Representations* (2023).

[340] R. Balestriero and Y. LeCun. "Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 26671–26685.

[341] P. H. Le-Khac, G. Healy, and A. F. Smeaton. "Contrastive representation learning: A framework and review". In: *Ieee Access* 8 (2020), pp. 193907–193934.

[342] L. Weng. "Contrastive Representation Learning". In: *lilianweng.github.io* (2021). URL: https://lilianweng.github.io/posts/2021-05-31-contrastive/.

[343] K. Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572.

[344] J. B. Kruskal. "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis". In: *Psychometrika* 29.1 (1964), pp. 1–27.

[345] L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[346] G. E. Hinton and S. Roweis. "Stochastic neighbor embedding". In: *Advances in neural information processing systems* 15 (2002).

[347] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[348] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[349] A. Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[350] X. Chen*, S. Xie*, and K. He. "An Empirical Study of Training Self-Supervised Vision Transformers". In: *arXiv preprint arXiv:2104.02057* (2021).

[351] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[352] J. Macqueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*. 1967.

[353] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22.

[354] J. Shi and J. Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905.

[355] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng. “Contrastive clustering”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 2021, pp. 8547–8555.

[356] W. V. Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. V. Gool. *SCAN: Learning to Classify Images without Labels*. 2020. arXiv: 2005.12320 [cs.CV]. URL: https://arxiv.org/abs/2005.12320.

[357] N. Adaloglou, F. Michels, H. Kalisch, and M. Kollmann. “Exploring the limits of deep image clustering using pretrained models”. In: *arXiv preprint arXiv:2303.17896* (2023).

[358] R. Grosse, R. R. Salakhutdinov, W. T. Freeman, and J. B. Tenenbaum. “Exploiting compositionality to explore a large space of model structures”. In: *arXiv preprint arXiv:1210.4856* (2012).

[359] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic. “On mutual information maximization for representation learning”. In: *arXiv preprint arXiv:1907.13625* (2019).

[360] P. Bachman, R. D. Hjelm, and W. Buchwalter. “Learning representations by maximizing mutual information across views”. In: *Advances in neural information processing systems* 32 (2019).

[361] F. Schroff, D. Kalenichenko, and J. Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[362] A. Bardes, J. Ponce, and Y. LeCun. “Vicreg: Variance-invariance-covariance regularization for self-supervised learning”. In: *arXiv preprint arXiv:2105.04906* (2021).

[363] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. “Supervised contrastive learning”. In: *Advances in neural information processing systems* 33 (2020), pp. 18661–18673.

[364] M. El Banani, K. Desai, and J. Johnson. “Learning visual representations via language-guided sampling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19208–19220.

[365] I. J. Good. “Maximum entropy for hypothesis formulation, especially for multidimensional contingency tables”. In: *The Annals of Mathematical Statistics* (1963), pp. 911–934.

[366] D. D. Baek, Z. Liu, R. Tyagi, and M. Tegmark. “Harmonic Loss Trains Interpretable AI Models”. In: *arXiv preprint arXiv:2502.01628* (2025).

[367] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[368] A. Ng, M. Jordan, and Y. Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems* 14 (2001).

[369] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka. *Debiased Contrastive Learning*. 2020. arXiv: 2007.00224 [cs.LG]. URL: https://arxiv.org/abs/2007.00224.

[370] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[371] W. Li, J. Xue, X. Zhang, H. Chen, Z. Chen, F. Huang, and Y. Cai. *Word-Graph2vec: An efficient word embedding approach on word co-occurrence graph using random walk technique*. 2023. arXiv: 2301.04312 [cs.CL]. URL: https://arxiv.org/abs/2301.04312.

[372] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea. "Graph Layouts by t-SNE". In: *Computer graphics forum*. Vol. 36. 3. Wiley Online Library. 2017, pp. 283–294.

[373] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV]. URL: https://arxiv.org/abs/2104.14294.

[374] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[375] E. Grave, A. Joulin, and Q. Berthet. *Unsupervised Alignment of Embeddings with Wasserstein Procrustes*. 2018. arXiv: 1805.11222 [cs.LG].

[376] D. Yan, L. Huang, and M. I. Jordan. "Fast approximate spectral clustering". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009, pp. 907–916.

[377] Y. Han and M. Filippone. "Mini-batch spectral clustering". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 3888–3895.

[378] P. Krähenbühl and V. Koltun. "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials". In: (2012). DOI: 10.48550/ARXIV.1210.5644. URL: https://arxiv.org/abs/1210.5644.

[379] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062* (2014).

[380] J. Ahn, S. Cho, and S. Kwak. "Weakly Supervised Learning of Instance Segmentation with Inter-pixel Relations". In: *CoRR* abs/1904.05044 (2019). arXiv: 1904.05044. URL: http://arxiv.org/abs/1904.05044.

[381] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[382] M. T. Teichmann and R. Cipolla. "Convolutional CRFs for semantic segmentation". In: *arXiv preprint arXiv:1805.04777* (2018).

[383] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. "Distributed representations of words and phrases and their compositionality". In: *arXiv preprint arXiv:1310.4546* (2013).

[384] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal. "graph2vec: Learning distributed representations of graphs". In: *arXiv preprint arXiv:1707.05005* (2017).

[385] O. Levy and Y. Goldberg. "Neural word embedding as implicit matrix factorization". In: *Advances in neural information processing systems* 27 (2014), pp. 2177–2185.

[386] X. Zhang, S. E. Chew, Z. Xu, and N. D. Cahill. "SLIC superpixels for efficient graph-based dimensionality reduction of hyperspectral imagery". In: *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*. Vol. 9472. International Society for Optics and Photonics. 2015, p. 947209.

[387] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[388] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. "Spatial transformer networks". In: *arXiv preprint arXiv:1506.02025* (2015).

[389] W. Falcon et al. "PyTorch Lightning". In: *GitHub. Note: https://github.com/PyTorchLightning/pytorch lightning* 3 (2019).

[390] A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. S. Kweon. *Learning to Localize Sound Source in Visual Scenes*. 2018. arXiv: 1803.03849 [cs.CV].

[391] T. Afouras, A. Owens, J. S. Chung, and A. Zisserman. *Self-Supervised Learning of Audio-Visual Objects from Video*. 2020. arXiv: 2008.04237 [cs.CV].

[392] W. Sun, J. Zhang, J. Wang, Z. Liu, Y. Zhong, T. Feng, Y. Guo, Y. Zhang, and N. Barnes. *Learning Audio-Visual Source Localization via False Negative Aware Contrastive Learning*. 2023. arXiv: 2303.11302 [cs.CV].

[393] S. Mo and P. Morgado. "A closer look at weakly-supervised audio-visual source localization". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 37524–37536.

[394] D. Snyder, G. Chen, and D. Povey. *MUSAN: A Music, Speech, and Noise Corpus*. 2015. arXiv: 1510.08484 [cs.SD].

[395] S. Marcel and Y. Rodriguez. "Torchvision the machine-vision package of torch". In: *Proceedings of the 18th ACM international conference on Multimedia*. 2010, pp. 1485–1488.

[396] K. He and Y. Wu. *MoCo: Momentum Contrast for Unsupervised Visual Representation Learning*. https://github.com/facebookresearch/moco. 2021.

[397] M. Ribeiro. *lime*. https://github.com/marcotcr/lime. 2021.

[398] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. *Slic superpixels*. Tech. rep. 2010.

[399] P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[400] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen. *Revisiting Training Strategies and Generalization Performance in Deep Metric Learning*. 2020. arXiv: 2002.08473 [cs.CV].

[401] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. “On variational bounds of mutual information”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5171–5180.

[402] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. “Variational inference: A review for statisticians”. In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[403] K. Crane, C. Weischedel, and M. Wardetzky. “Geodesics in heat: A new approach to computing distance based on heat flow”. In: *ACM Transactions on Graphics (TOG)* 32.5 (2013), pp. 1–11.