November 2023

Learning to See with Minimal Human Supervision

Zezhou Cheng
University of Massachusetts Amherst

LEARNING TO SEE WITH MINIMAL HUMAN SUPERVISION

A Dissertation Presented

by

ZEZHOU CHENG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2023

Robert and Donna Manning College of
Information and Computer Sciences

LEARNING TO SEE WITH MINIMAL HUMAN SUPERVISION

A Dissertation Presented

by

ZEZHOU CHENG

Approved as to style and content by:

Subhransu Maji, Chair

Daniel Sheldon, Member

Erik Learned-Miller, Member

Varun Jampani, Member

Ramesh K. Sitaraman, Associate Dean for
Educational Programs and Teaching,
Robert and Donna Manning College of
Information and Computer Sciences

ACKNOWLEDGMENTS

First, I would like to thank Subhransu Maji for being an amazing advisor! Throughout my six-year Ph.D. journey, his guidance and support have been invaluable. The joy I experienced during this time as a Ph.D. student and my aspiration to become a faculty, devoting my life to computer vision research, is a testament to his inspiration. I would like to thank my co-advisor Daniel Sheldon for his mentoring through this journey. His passion for ecological research has continually motivated me to broaden my horizons beyond computer vision, and he has perpetually served as an exemplary figure in my teaching endeavors. I would like to express my sincere appreciation to Erik Learned-Miller, who has created a joyful atmosphere in the vision lab. His enlightening insights and humor have consistently been my prime motivation to attend every single group meeting in the past six years. I would also like to thank Varun Jampani for making my time at Google fruitful and providing thoughtful feedback on this dissertation. Additionally, my deepest thanks go to Georgia Gkioxari for offering me an incredible opportunity to explore my research interests in 3D vision at Caltech.

Next, I'd like to extend heartfelt thanks to Huaizu Jiang and Rui Wang for their unwavering support throughout my academic career. I'm also grateful to my internship mentors Menglei Chai and Sergey Tulyakov at Snap Research, and Ameesh Makadia and Carlos Esteves at Google Research, who offered me fantastic opportunities and mentorship, enabling me to delve into industry research. I want to express my appreciation for my collaborators Matheus Gadelha, Gustavo Perez, Jong-Chyi Su, Oindrila Saha, Wenlong Zhao, Jian Ren, Hsin-Ying Lee, Kyle Olszewski, Zeng Huang, Abhishek Kar, Maria Belotti, Yuting Deng, Kyle Horton, Yachan Liu, and Peng Bai, with whom I've had the privilege to work on exciting projects. A big thanks

to my lab mates Tsung-Yu Lin, Hang Su, Souyoung Jin, Gopal Sharma, Chenyun Wu, Zitian Chen, Ashish Singh, Aaron Sun, Rangel Daroya, Max Hamilton, Difan Liu, Zhan Xu, Deep Chakraborty, Fabien Delattre, Zhipeng Tang, David Dirnfeld, Yang Zhou, Dmitry Petro, Pratheba Selvaraju, Vikas Thamizharasan for creating such an inspiring lab environment. I also want to thank my dear friends Pengshan Cai, Xiang Li, Dongxu Zhang, Zhichao Yang, Mengxue Zhang, Zhiqi Huang, Puxuan Yu, Bo Guan, and Anqi He who have filled my time in Amherst with joy and companionship.

To conclude, I wish to express my profound gratitude towards my parents and my endearing little sister. Their unwavering faith and love have perpetually served as my greatest source of motivation, bolstering me through the most challenging times. The completion of the research work presented in this dissertation would have been impossible without their support. I owe a heartfelt thank you to my father for his unstinting support towards my education and my professional career, to my mother for her tireless care for our family, and to my younger sister, whose presence has brought immense joy and love to our family life.

ABSTRACT

LEARNING TO SEE WITH MINIMAL HUMAN SUPERVISION

SEPTEMBER 2023

ZEZHOU CHENG

PH.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Subhransu Maji

Deep learning has significantly advanced computer vision in the past decade, paving the way for practical applications such as facial recognition and autonomous driving. However, current techniques depend heavily on human supervision, limiting their broader deployment. This dissertation tackles this problem by introducing algorithms and theories to minimize human supervision in three key areas: data, annotations, and neural network architectures, in the context of various visual understanding tasks such as object detection, image restoration, and 3D generation.

First, we present self-supervised learning algorithms to handle in-the-wild images and videos that traditionally require time-consuming manual curation and labeling. We demonstrate that when a deep network is trained to be invariant to geometric and photometric transformations, representations from its intermediate layers are highly predictive of object semantic parts such as eyes and noses. This insight offers a simple unsupervised learning framework that significantly improves the efficiency and accuracy of few-shot landmark prediction and matching. We then present a technique for learning single-view 3D object pose estimation models by utilizing in-the-wild videos

where objects turn (*e.g.*, cars in roundabouts). This technique achieves competitive performance with respect to existing state-of-the-art without requiring any manual labels during training. We also contribute an Accidental Turntables Dataset, containing a challenging set of 41,212 images of cars in cluttered backgrounds, motion blur, and illumination changes that serve as a benchmark for 3D pose estimation.

Second, we address variations in labeling styles across different annotators, which leads to a type of noisy label referred to as *heterogeneous label*. This variability in human annotation can cause subpar performance during both the training and testing phases. To mitigate this, we have developed a framework that models the labeling styles of individual annotators, reducing the impact of human annotation variations and enhancing the performance of standard object detection models. We have also applied this framework to analyze ecological data, which are often collected opportunistically across different case studies without consistent annotation guidelines. Through this application, we have obtained several insightful observations into large-scale bird migration behaviors and their relationship to climate change.

Our next study explores the challenges of designing neural networks, an area that lacks a comprehensive theoretical understanding. By linking deep neural networks with Gaussian processes, we propose a novel Bayesian interpretation of the deep image prior, which parameterizes a natural image as the output of a convolutional network with random parameters and random input. This approach offers valuable insights to optimize the design of neural networks for various image restoration tasks.

Lastly, we introduce several machine-learning techniques to reconstruct and edit 3D shapes from 2D images with minimal human effort. We first present a generic multi-modal generative model that bridges 2D images and 3D shapes via a shared latent space, and demonstrate its applications on versatile 3D shape generation and manipulation tasks. Additionally, we develop a framework for joint estimation of 3D neural scene representation and camera poses. This approach outperforms prior

works and allows us to operate in the general SE(3) camera pose setting, unlike the baselines. The results also indicate this method can be complementary to classical structure-from-motion (SfM) pipelines as it compares favorably to SfM on low-texture and low-resolution images.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

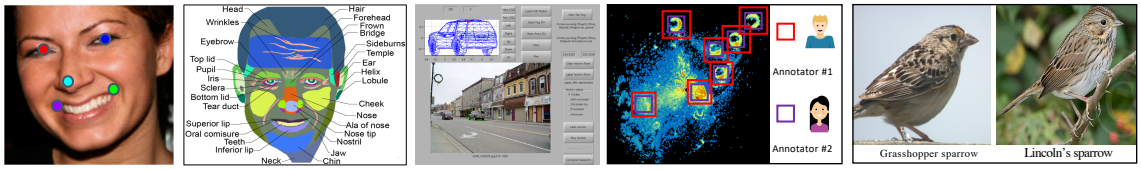
INTRODUCTION

Over the past decade, the field of computer vision has made remarkable progress thanks to deep learning techniques. These techniques typically involve a simple recipe: collecting source data, annotating data, and designing neural network architectures. However, each step requires significant human effort, which has limited the adoption and performance of deep learning in novel and complex visual reasoning tasks.

While there is an abundance of images and videos available on the internet, platforms like Instagram and YouTube host mostly unlabeled data, and labeling all of it is infeasible. Moreover, the distribution of objects in the real world is often long-tailed. Benchmarks commonly used in computer vision research, such as ImageNet [53], are carefully curated to be well-balanced, diversified, and human-annotated. This curation process demands considerable human resources.

Most importantly, the gap between these benchmarks and real-world data raises concerns about the generalizability of research findings to real-world applications. The substantial human effort involved in data curation and the potential disconnect between curated benchmarks and real-world scenarios highlight the need for more efficient and generalizable deep learning techniques.

The cost of collecting human annotations is a significant barrier in many vision tasks. For example, annotating the landmarks or semantic parts of an object is much more time-consuming than categorizing the image (Fig. 1.1a,b); annotating the 3D pose of an object is often done by reasoning with 3D model's projection to the 2D image (Fig. 1.1c); annotating objects with fine-grained labels (*e.g.*, Grasshopper



(a) Landmark detection (b) Part segmentation (c) 3D pose estimation (d) Object detection (e) Fine-grained recognition

Figure 1.1: Visual understanding tasks. For these tasks, human annotations are time-consuming (a-c), noisy due to the labeling variation across different annotations (d), and even require domain-specific expertise (e). (c) presents the pose annotation interface from the PASCAL3D+ dataset [260]. In (d), annotators are asked to label the swallow roosts (*i.e.*, ring-like patterns) in weather radar data.

sparrow vs. Lincoln's sparrow) requires strong domain-specific expertise (Fig. 1.1e). In addition, labeling without clearly defined protocols leads to a variation in labeling styles of different annotators (*i.e.*, heterogeneous labels), which can make subsequent learning harder (*e.g.*, noise in bounding box annotations (Fig. 1.1d)).

Designing neural network architectures also demands substantial human supervision. Over the past decade, numerous diverse neural networks have been developed for computer vision tasks (*e.g.*, VGG [203], ResNet [96]), but our understanding of these networks relies heavily on empirical studies. Designing a neural network for a novel visual reasoning task typically involves an expensive, time-consuming trial-and-error process. Therefore, understanding deep models theoretically is essential for designing neural network architectures more effectively.

In addition to the substantial costs associated with obtaining high-quality human annotations and designing neural networks, data collection itself can be a costly endeavor. This is particularly true for the creation and modification of 3D objects, which demands significant human effort and expertise compared to working with 2D images or videos. Given these challenges, the reconstruction and manipulation of 3D assets from 2D images have emerged as a focal problem within the field of computer vision. This is evidenced by decades of dedicated research into Structure-from-Motion (SfM) algorithms [90]. However, SfM algorithms often falter when dealing with sparse 2D observations and surfaces with low texture, leading to a significant decrease in per-

formance. Furthermore, SfM, when combined with multi-view stereo (MVS), only offers a rudimentary description of 3D geometry and texture. As a result, there is considerable scope for enhancing the efficacy of current techniques within this domain.

In this dissertation, we explore four research questions: (1) how can we learn from in-the-wild images or videos with minimal data curation and labeling? (2) how can we learn from heterogeneous labels induced by variations in labeling styles amongst different annotators? (3) how can we theoretically understand neural network architectures? (4) how can we reconstruct and edit 3D shapes with minimal human effort? To address these questions, we develop unsupervised learning algorithms for various vision tasks [38, 225]; we provide a general machine learning framework to learn from annotations with different labeling styles and demonstrate its application in a series of ecological studies [17, 55, 174, 224]; we introduce a theoretical understanding of deep convolutional neural networks through the lens of Gaussian processes [226]; and we propose several techniques to reconstruct and manipulate 3D shapes from 2D images [222, 223].

1.1 Contributions

In Chapter 3, we introduce a simple and effective self-supervised learning approach by combining instance-discriminative and spatially-discriminative contrastive learning. We show that the proposed approach surpasses prior state-of-the-art on few-shot landmark prediction and landmark matching tasks [38]. Next, we propose to learn single-view 3D object pose estimation models by utilizing a new source of data — in-the-wild videos where objects turn [225]. We also contribute an *Accidental Turntables Dataset* which serves as a challenging benchmark for 3D pose estimation. Our technique achieves competitive performance with existing state-of-the-art on standard benchmarks without requiring any pose labels during training.

List of publications related to Chapter 3:

- [38] **Cheng, Z.**, Su, J. C., & Maji, S. (2021). *On equivariant and invariant learning of object landmark representations.* In Proceedings of the IEEE/CVF International Conference on Computer Vision.
- [225] **Cheng, Z.**, Gadelha, M., & Maji, S. (2022). *Accidental Turntables: learning 3D pose by watching objects turn.* In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop.

In Chapter 4, we introduce an EM framework for learning from heterogeneous labels [224] (Fig. 1.1d). We apply this framework to build an integrated system for large-scale ecological studies — detecting and tracking communal birds in weather radar data [224]. This system has provided biologists insights about the migration behavior of birds in relation to environmental change and yielded many insightful ecological findings [17, 55, 174].

List of publications related to Chapter 4:

- [224] **Cheng, Z.**, Gabriel, S., Bhambhani, P., Sheldon, D., Maji, S., Laughlin, A., & Winkler, D. (2020, April). *Detecting and tracking communal bird roosts in weather radar data.* In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 01, pp. 378-385).
- [174] Pérez, G., Zhao, W., **Cheng, Z.**, Belotti, M., Deng, Y., Simons, V., Tielens, E., Kelly, J., Horton, K., Maji, S., Sheldon, D. *Using spatiotemporal information in weather radar data to detect and track communal bird roosts.* BioRxiv, 2022, doi: https://doi.org/10.1101/2022.10.28.513761.
- [55] Deng, Y., Belotti, M., Zhao, W., **Cheng, Z.**, Pérez, G., Tielens, E., Simons, V., Sheldon, D., Maji, S., Kelly, J., Horton, K. *Quantifying long-term phenological patterns of aerial insectivores roosting in the Great Lakes region using weather surveillance radar.* Global Change Biology, 2022, 00, 1– 13. doi.org/10.1111/gcb.16509.
- [17] Belotti, M., Deng, Y., Zhao, W., Simons, V., **Cheng, Z.**, Pérez, G., Tielens, E., Maji, S., Sheldon, D., Kelly, J., Horton, K. *Long-term analysis of persistence and size of swallow and*

martin roosts in the US Great Lakes. Remote Sensing in Ecology and Conservation, 2023; doi.org/10.1002/rse2.323.

In Chapter 5, we provide a theoretical analysis of the inductive bias of a random CNN in the context of unsupervised image restoration. We show that the Deep Image Prior (DIP) [237] is asymptotically equivalent to a Gaussian process (GP) as the network width goes to infinity, and we avoid the need for early stopping using stochastic gradient Langevin dynamics (SGLD) [247] for unsupervised image restoration tasks.

List of publications related to Chapter 5:

- [226] **Cheng, Z.**, Gadelha, M., Maji, S., & Sheldon, D. (2019). *A bayesian perspective on the deep image prior.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

In Chapter 6, we introduce a generic multi-modal generative model that couples the 2D modalities (*e.g.*, natural images or sketches) and implicit 3D representations (*e.g.*, signed distance functions) through shared latent spaces, enabling versatile 3D generation and manipulation tasks. We also present a local-to-global framework that jointly estimates 3D neural scene representation [145] and camera poses, and demonstrates its complementary performance to classical Structure-from-Motion algorithms.

List of publications related to Chapter 6:

- [222] **Cheng, Z.**, Chai, M., Ren, J., Lee, H. Y., Olszewski, K., Huang, Z., ... & Tulyakov, S. (2022). Cross-modal 3d shape generation and manipulation. In European Conference on Computer Vision.
- [223] **Cheng, Z.**, Esteves, C., Jampani, V., Kar, A., Maji, S., & Makadia, A. (2023). LU-NeRF: Scene and Pose Estimation by Synchronizing Local Unposed NeRFs. In Proceedings of the IEEE/CVF International Conference on Computer Vision.

Finally, future works will focus on three essential areas to advance the development of intelligent agents capable of understanding and interacting with the 3D visual world: holistic 3D scene understanding and reconstruction, multi-modal perception,

and learning from real-world data distributions. I will also strengthen collaborations with researchers from the industry, graphics, robotics, ecology, and other scientific fields to apply AI techniques to solve real-world problems and promote the practical application of AI across various domains.

Source codes and datasets. In addition to publishing my research in conferences and journals, I have actively contributed to the open-source community. The datasets and codes developed or collected in this dissertation can be accessed on GitHub (`https://github.com/cvl-umass`), which is maintained by the members of the computer vision lab at UMass Amherst.

CHAPTER 2

BACKGROUND & LITERATURE REVIEW

In this chapter, we provide the background information related to several research topics covered in this dissertation. We begin by reviewing unsupervised learning algorithms that exploit unlabeled images or videos in Section 2.1. We then explore previous work on learning from noisy labels in Section 2.2. We discuss the literature on designing and theoretically understanding neural network architectures in Section 2.3. Lastly, we illustrate existing works on 3D reconstruction and manipulation in Section 2.4. We present additional related works that are specific to only certain chapters later.

2.1 Learning from unlabeled data

The main breakthrough in computer vision in the past decade is achieved by supervised learning which heavily relies on costly human annotations. This is manifested in the huge efforts in the community to collect a large corpus of image datasets with detailed annotations such as ImageNet [53] and MS-COCO [127]. However, supervised learning is not scalable, and the natural learning of our humans is largely unsupervised. For these reasons, unsupervised learning that exploits unlabeled data has gained great attention in the deep learning era.

The goal of unsupervised learning is to discover patterns or structures within the data without explicit labels. Common unsupervised learning techniques include clustering which groups similar instances and dimensionality reduction which projects the data into a low dimensional space. In particular, unsupervised (or self-supervised)

representation learning learns data representations that are useful for downstream tasks, without explicit human supervision. A wide range of self-supervised representation learning methods trains a neural network to recover the missing information such as colorization [279] and inpainting [171] or predict the spatial context such as rotation prediction [73] and jigsaw puzzle [159]. More recent works show that the unsupervised contrastive learning methods [10, 34, 35, 36, 60, 93, 100, 163, 232, 257] outperforms algorithms based on aforementioned pretext tasks. These contrastive learning objectives [86] are often expressed in terms of noise-contrastive estimation (NCE) [85] (or maximizing mutual information [100, 163]) between different views obtained by geometrically and photometrically transforming an image. The learned representations thus encode invariances to these transformations while preserving information relevant to downstream tasks.

However, the effectiveness of unsupervised learning depends on how well these invariances relate to those desired for end tasks. Despite recent advances, existing methods for unsupervised learning significantly lack in comparison to their supervised counterparts in the few-shot setting [79]. Moreover, their effectiveness for detailed visual understanding tasks (*e.g.*, landmark detection) has not been sufficiently studied in the literature. This motivates us to explore unsupervised learning to understand object structure and pose described in Chapter 3.

2.2 Learning from noisy annotations

Despite the rapid progress in unsupervised learning, supervised learning remains the dominant method in practical applications due to its superior and robust performance. However, large annotated datasets often suffer from noisy or incorrect labels, which, if not handled appropriately, can significantly degrade the performance of machine learning models [157]. The typical source of noisy labels includes ambiguous image content, poor image quality, or labeling mistakes by annotators. Var-

ious methods have been proposed to learn models from noisy annotations, ranging from noise-robust loss functions [72, 241] to data cleaning [24] and bootstrapping [146, 150, 172, 219, 264]. Techniques that explicitly model the noise process have also been proposed in the literature [146, 172]. Despite the vast literature on this topic, most prior works are designed for the classification task.

This dissertation addresses learning from a specific type of noisy labels which are induced by the variations in the labeling style across different annotators in the object detection task, *i.e.*, heterogeneous labels. Related to our work, Jiang *et al.* [107] discuss how systematic differences in labeling style across face-detection benchmarks significantly complicate evaluation, and proposes fine-tuning techniques for style adaptation. Differently, we propose a principled method that explicitly models the annotators' labeling style in Chapter 4.

2.3 Neural network architectures and Gaussian process

Designing neural network architectures has been one of the major research topics in the deep learning era. Extensive variants of network architectures have been proposed in the past decade. Seminar works include AlexNet [116], VGG [204], Inception [216], ResNet [96], U-Net [180], and Vision Transformer [59]. Despite the vast work in this field, designing neural networks remains an art instead of a science — it usually requires extensive trial and error to build a suitable network architecture for a novel task. Efforts have been made to theoretically understand neural network architectures, with the target of designing the network principally. Here we briefly describe prior works that interpret the neural network architectures through the lens of Gaussian processes.

Back in 1995, Neal [156] showed that a two-layer network converges to a Gaussian process as its width goes to infinity. Later, Williams [251] provided expressions for the covariance function of networks with Sigmoid and Gaussian transfer func-

tions. Cho and Saul [41] presented kernels for the ReLU and the Heaviside step non-linearities and investigated their effectiveness with kernel machines. More recently, several works [119, 144] have extended these results to deep networks and derived covariance functions for the resulting GPs. Similar analyses have also been applied to convolutional networks. Garriaga-Alonso *et al.* [69] investigated the GP behavior of convolutional networks with residual layers while Borovykh [21] analyzed the covariance functions in the limit when the filter width approaches infinity. Novak *et al.* [160] evaluated the effect of pooling layers in the resulting GP. Much of this work has been applied to prediction tasks, where given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, a covariance function induced by a deep network is used to estimate the posterior $p(y|x, \mathcal{D})$ using standard GP machinery. Similar to these works, we make a connection between the neural network architectures with the Gaussian process, however, the network architectures and the tasks are both different from prior works, as described in Chapter 5.

2.4 3D reconstruction and manipulation

Structure from Motion (SfM). Jointly recovering 3D scenes and estimating camera poses from multiple views of a scene is a classic problem in computer vision [91]. Numerous techniques have been proposed for SfM [167, 196] with unordered image collections and visual-SLAM for sequential data [153, 218]. These techniques are largely built upon local features [56, 139, 184, 213] and require accurate detection and matching across images. The success of these techniques has led to their widespread adoption, and existing deep-learning approaches for scene representation and novel view synthesis are designed with the implicit assumption that the SfM techniques provide accurate poses in the wild. For example, NeRF [149] and its many successors (*e.g.*, [12, 13, 152]) utilize poses estimated offline with COLMAP [130, 196]. However, COLMAP can fail on textureless regions and low-resolution images. Chapter 6 in-

troduces a novel approach that jointly estimates 3D scenes and camera poses, which shows complementary performance to SfM.

Machine learning for 3D reconstruction. Extensive works have explored the problem of 3D reconstruction from different modalities, such as RGB images [43, 109], videos [271], sketches [83, 108, 283, 288], or even text [32], using machine learning techniques. This problem has also been explored under diverse representations [37, 43, 64, 74, 133, 147, 170, 205, 244, 269] and different levels of supervision [43, 74, 75, 109, 271]. Despite the diverse settings of this problem, the encoder-decoder network, which maps the source modalities to 3D shape directly in a feed-forward manner, remains the most popular 3D reconstruction model [43, 109, 170, 244]. However, such feed-forward networks are not robust to input domain shift (*e.g.*, incomplete data). We introduce a generic multi-model generative model to tackle this issue in Chapter 6.

Shape and appearance manipulation. Numerous interactive tools have been developed for image editing [81, 120, 122, 124, 183, 282] and 3D shape manipulations [5, 51, 173, 193]. More recently, generative modeling of natural images [77, 207] has became a "Swiss knife" for image editing problems [1, 14, 15, 82, 168, 187, 199, 200, 291]. Novel interactive tools have also been proposed recently to edit implicit 3D representations [149, 170]. For example, DualSDF [87] edits the SDFs [170] via shape primitives (*e.g.*, spheres). Sketch2Mesh [83] reconstructs shapes from sketch with an encoder-decoder network and refines 3D shapes via differentiable rendering. EditNeRF [134] edits the radiance field [149] by fine-tuning the network weights based on the user's scribbles.

CHAPTER 3

LEARNING FROM UNLABELED IMAGES AND VIDEOS

Understanding object pose and its structure is a long-standing computer vision task with wide applications in practice, such as monitoring animal behaviors or human activities. The object pose can be annotated in various forms, depending on the object categories. For rigid objects (*e.g.*, cars), the pose is usually represented by 3D rotation and translation, while the pose of non-rigid objects is typically characterized by 2D landmarks (*e.g.*, eyes, noses). Undoubtedly, the manual annotation process is labor-intensive and prone to unavoidable human annotation errors. To tackle this issue, we present a self-supervised representation learning approach for learning landmark representation from unlabeled images in Section 3.1. We also develop a novel unsupervised method for training 3D pose estimation models from unlabeled videos in Section 3.2.

3.1 Learning landmark representations from unlabeled images

Given a collection of images, humans are able to discover landmarks by modeling the shared geometric structure across instances. This idea of geometric equivariance has been widely used for the unsupervised discovery of object landmark representations. In this work, we develop a simple and effective approach by combining instance-discriminative and spatially-discriminative contrastive learning. We show that when a deep network is trained to be invariant to geometric and photometric transformations, representations emerge from its intermediate layers that are highly predictive of object landmarks. Stacking these across layers in a "hypercolumn" and project-

ing them using spatially-contrastive learning further improves their performance on matching and few-shot landmark regression tasks. We also present a unified view of existing equivariant and invariant representation learning approaches through the lens of contrastive learning, shedding light on the nature of the invariances learned. Experiments on standard benchmarks for landmark learning, as well as a new challenging one we propose, show that the proposed approach surpasses prior state-of-the-art.

3.1.1 What is landmark?

The term "landmark" does not hold a consistent definition across various literature. Many previous studies, particularly in the domain of supervised learning, define landmarks as points of interest within an image that are manually annotated or chosen by humans [66, 262, 286, 287]. These landmarks, commonly known as *human-defined* landmarks, are denoted using 2D pixel coordinates and typically represent identifiable parts of objects, such as the eyes and noses of human or animal faces.

However, a broader definition of landmarks exists, one that is founded on the principles of equivariance and invariance [229, 230, 231]. Primarily, a landmark should exhibit equivariance toward image transformations. A representation $\Phi : \mathcal{X} \to \mathbb{R}^C$ is considered to be equivariant or covariant with respect to a transformation g for input $\mathbf{x} \in \mathcal{X}$ if a corresponding map $M_g : \mathbb{R}^C \to \mathbb{R}^C$ exists, satisfying the condition: $\forall \mathbf{x} \in \mathcal{X} : \Phi(g\mathbf{x}) \approx M_g \Phi(\mathbf{x})$. This condition essentially states that the representation should transform predictably in response to the input transformation.

These transformations in natural images could take various forms - geometric (such as translation, scaling, and rotation), photometric (like color changes), or even more complex transformations (including occlusion, viewpoint, or instance variations). Computer vision boasts a long history of designing covariant representations, exemplified by techniques such as SIFT [138]. Contrary to these classical equivariant descriptors, a landmark should display invariance to intra-category variations, which

include the variation in eye shapes across different identities. In our study, we utilize this more general definition of landmarks and aim at learning landmark representations that fulfill the outlined equivariant and invariant properties, which can be used to establish correspondences across objects and to predict landmarks such as eyes and noses when provided with a few labeled examples.

3.1.2 Overview

In the previous section, we define landmarks based on the property of equivariance. This leads to a natural method for discovering landmarks, namely to learn a representation that geometrically transforms in the same way as the object, a property known as *geometric equivariance* (Fig. 3.1a) [229, 230, 231]. However, useful invariances may not be learned (*e.g.*, the raw pixel representation itself is equivariant), limiting their applicability in the presence of clutter, occlusion, and inter-image variations.

A different line of work has proposed instance discriminative *contrastive learning* as an unsupervised objective [10, 34, 60, 86, 93, 98, 100, 163, 232, 257, 293]. The goal is to learn a representation Φ that has higher similarity between an image $\mathbf{x}$ and its transformation $\mathbf{x}'$ than with a different one $\mathbf{z}$, $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle \gg \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$, as illustrated in Fig. 3.1b. A combination of geometric (*e.g.*, cropping and scaling) and photometric (*e.g.*, color jittering and blurring) transformations are used to encourage the representation to be *invariant* to these transformations while being *distinctive* across images. Recent work [34, 35, 36, 93] has shown that contrastive learning is effective, even outperforming ImageNet [53] pre-training on various tasks. However, to predict landmarks a representation cannot be invariant to geometric transformations. This work asks the question: *are equivariant losses necessary for unsupervised landmark discovery?* In particular, do representations predictive of object landmarks automatically emerge in intermediate layers of a deep network trained to be invariant

to image transformations? While empirical evidence suggests that semantic parts emerge when deep networks are trained on supervised tasks [76, 289], is it also the case for unsupervised learning?

This work aims to address these by presenting a unified view of the equivariant and invariant learning approaches. We show that when a deep network is trained to be invariant to geometric and photometric transformations, its intermediate-layer representations are highly predictive of landmarks (Fig. 3.1b). The emergence of invariance and the loss of geometric equivariance is gradual in the representation hierarchy, a phenomenon that has been studied empirically [121, 277] and theoretically [2, 233, 234]. This observation motivates a *hypercolumn* representation [88], which we find to be more effective for landmark predictions (Fig. 3.1c).

We also observe that objectives used in equivariant learning can be seen as a contrastive loss between representations across locations within the *same image*, as opposed to invariant learning where the loss is applied *across images* (Fig. 3.1). This observation sheds light on the nature of the invariances learned by the two approaches. It also allows us to obtain a compact representation of the high-dimensional hypercolumns simply by learning a linear projection under the spatially contrastive objective. The projection results in spatially distinctive representations and significantly improves the landmark matching performance (Tab. 3.1 and Fig. 3.2).

To validate these claims, we perform experiments by training deep networks using Momentum Contrast (MoCo) [93] on several landmark matching and detection benchmarks. Other than commonly used ones, we also present a comparison by learning on a challenging dataset of birds from the iNaturalist dataset [240] and evaluating on the CUB dataset [242]. We show that the contrastive-learned representations (without supervised regression) can be predictive in landmark matching experiments. For landmark detection, we adapt the commonly used linear evaluation setting by varying the number of labeled examples (Fig. 3.3 & 3.4). Our approach is simple, yet it offers con-

sistent improvements over prior approaches [105, 229, 230, 231, 284] (Tab. 3.2). While the hypercolumn representation leads to a larger embedding dimension, it comes at a modest cost as our approach outperforms the prior state-of-the-art [229], with as few as 50 annotated training examples on the AFLW benchmark [115] (Fig. 3.4). Furthermore, we use dimensionality reduction based on the equivariant learning to improve the performance on landmark matching (Tab. 3.1), as well as landmark prediction in the low data regime (Tab. 3.4).

3.1.3 Related works

Deep representations. Invariance and equivariance in deep network representations result from both the architecture (*e.g.*, convolutions lead to translational equivariance while pooling leads to translational invariance) and learning (*e.g.*, invariance to categorical variations). Lenc *et al.* [121] showed that early-layer representations of a deep network are nearly equivariant as they can be "inverted" to recover the input, while later layers are more invariant. Similar observations have been made by visualizing these representations [140, 277]. The gradual emergence of invariance can also be theoretically understood in terms of a "information bottleneck" in the feed-forward hierarchy [2, 233, 234]. While equivariance to geometric transformations is relevant for landmark representations, the notion can be generalized to other transformation groups [46, 71].

Landmark discovery. Empirical evidence [164, 289] suggests that semantic parts emerge when deep networks are trained on supervised tasks. This has inspired architectures for image classification that encourage part-based reasoning, such as those based on texture representations [6, 45, 128] or spatial attention [67, 198, 263]. In contrast, our work shows that *parts also emerge when models are trained in an unsupervised manner.* When no labels are available, equivariance to geometric transformations provides a natural self-supervisory signal. The equivariance constraint requires

$\Phi_u(\mathbf{x})$, the representation of $\mathbf{x}$ at location u, to be invariant to the geometric transformation g of the image, *i.e.*, $\forall \mathbf{x}, u : \Phi_{gu}(g\mathbf{x}) = \Phi_u(\mathbf{x})$ (Fig. 6.1a). This alone is not sufficient since both $\Phi_u(\mathbf{x}) = \mathbf{x}_u$ and $\Phi_u(\mathbf{x}) =$ constant satisfy this property. Constraints based on locality [229, 230] and diversity [231] have been proposed to avoid this pathology. Yet, inter-image invariance is not directly enforced. Another line of work is based on a generative modeling approach [18, 105, 106, 137, 189, 202, 249, 270, 284]. These methods implicitly incorporate equivariant constraints by modeling objects as deformation (or flow) of a shape template together with appearance variation in a disentangled manner.

3.1.4 Approach

Let $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ denote an image of an object, and $u \in \Omega = \{0, \ldots, H-1\} \times \{0, \ldots, W-1\}$ denote pixel coordinates. The goal is to learn a function $\Phi_u(\mathbf{x}) : \Omega \to \mathbb{R}^C$ that outputs a pixel representation at spatial location u of input $\mathbf{x}$ that is predictive for object landmarks. We assume $C \gg 3$ aiming to learn a high-dimensional representation of landmarks. This is similar to [229] which learns a local descriptor for each landmark, and unlike those that represent them as a discrete set [287], or on a planar ($C = 2$) [231, 284] or spherical ($C = 3$) [230] coordinate system. In other words the representation should be predictive of landmarks or effective for matching, without requiring compactness or topology in the embedding space. Note that this is in contrast to some work on literature where a fixed set of landmarks are discovered (*e.g.*, [105, 231, 284]). One may obtain this, for instance, by clustering the landmark representations in the embedding space.

We describe commonly used equivariance constraints for unsupervised landmark discovery [229, 230, 231], followed by models based on invariant learning [93, 163]. We then present our approach that integrates the equivariant and invariant learning approaches.

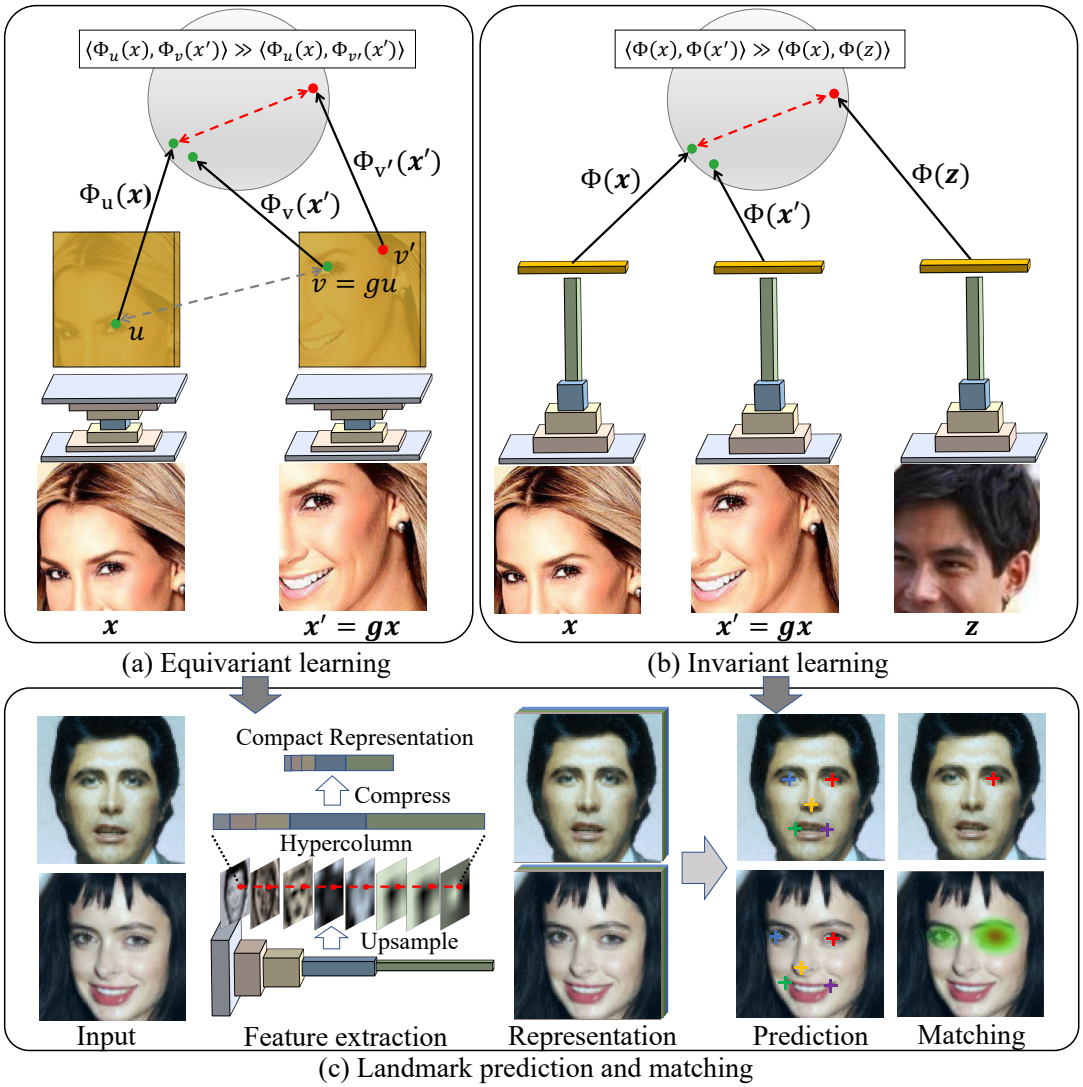


Figure 3.1: Equivariant and invariant learning. (a) Equivariant learning requires representations across locations to be invariant to a geometric transformation g while being distinctive across locations. (b) Invariant learning encourages the representations to be invariant to transformations while being distinctive across images. Thus both can be seen as instances of contrastive learning. (c) A hypercolumn feature and its compact representation are highly predictive of object landmarks.

Equivariant learning. The equivariance constraint requires $\Phi_u(\mathbf{x})$, the representation of $\mathbf{x}$ at location u, to be invariant to the geometric deformation of the image (Fig. 3.1a). Given a geometric warping function $g : \Omega \to \Omega$, the representation of $\mathbf{x}$ at u should be same as the representation of the transformed image $\mathbf{x}' = g\mathbf{x}$ at $v = gu$, that is, $\forall \mathbf{x}, u \in \Omega : \Phi_v(\mathbf{x}') = \Phi_u(\mathbf{x})$. This constraint can be captured by the loss:

$$\mathcal{L}_{equi} = \frac{1}{|\Omega|} \sum_{u \in \Omega} \|\Phi_u(\mathbf{x}) - \Phi_v(\mathbf{x}')\|^2. \tag{3.1}$$

A diversity (or locality) constraint is necessary to encourage the representation to be distinctive across locations. For example, Thewlis *et al.* [230] proposed the following:

$$\mathcal{L}_{div} = \frac{1}{|\Omega|} \sum_{u \in \Omega} \|gu - \operatorname*{argmax}_v \langle \Phi_u(\mathbf{x}), \Phi_v(\mathbf{x}') \rangle\|^2, \tag{3.2}$$

which they replaced by a probabilistic version that combines both the losses as:

$$\mathcal{L}'_{equi} = \frac{1}{|\Omega|^2} \sum_{u \in \Omega} \sum_{v \in \Omega} \|gu - v\| \; p(v|u; \Phi, \mathbf{x}, \mathbf{x}'). \tag{3.3}$$

Here $p(v|u; \Phi, \mathbf{x}, \mathbf{x}')$ is the probability of pixel u in image $\mathbf{x}$ matching v in image $\mathbf{x}'$ with Φ as the encoder shared by $\mathbf{x}$ and $\mathbf{x}'$ computed as below, and $\tau \in \mathbb{R}^+$ is a scale parameter:

$$p(v|u; \Phi, \mathbf{x}, \mathbf{x}') = \frac{\exp(\langle \Phi_u(\mathbf{x}), \Phi_v(\mathbf{x}') \rangle / \tau)}{\sum_{t \in \Omega} \exp(\langle \Phi_u(\mathbf{x}), \Phi_t(\mathbf{x}') \rangle / \tau)}. \tag{3.4}$$

Invariant learning. Contrastive learning is based on the similarity over pairs of inputs (Fig. 3.1b). Given an image $\mathbf{x}$ and its transformation $\mathbf{x}'$ as well as other images $\mathbf{z}_i$, $i \in \{1, 2...N\}$, the InfoNCE [163] loss minimizes:

$$\mathcal{L}_{inv} = -\log \frac{\exp\left(\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle\right)}{\sum_{i=1}^{N} \exp(\langle \Phi(\mathbf{x}), \Phi(\mathbf{z}_i) \rangle)}. \tag{3.5}$$

The objective encourages representations to be invariant to transformations while being distinctive across images. To address the computational bottleneck in evaluating

the denominator, Momentum Contrast (MoCo) [93] computes the loss over negative examples using a dictionary queue and updates the parameters based on momentum.

Transformations. The space of transformations used to generate image pairs $(\mathbf{x}, \mathbf{x}')$ plays an important role in learning. A common approach is to apply a combination of *geometric transformations*, such as cropping, resizing, and thin-plate spline warping, as well as *photometric transformations*, such as color jittering and adding JPEG noise. Transformations can also denote channels of an image or modalities such as depth and color [232].

Hypercolumns. A deep network of n layers (or blocks[1]) can be written as $\Phi(\mathbf{x}) = \Phi^{(n)} \circ \Phi^{(n-1)} \circ \cdots \circ \Phi^{(1)}(\mathbf{x})$. A representation $\Phi(\mathbf{x})$ of size $H' \times W' \times C$ can be spatially interpolated to the input size $H \times W \times C$ to produce a pixel representation $\Phi_u(\mathbf{x}) \in \mathbb{R}^C$. The hypercolumn representation of layers $k_1, k_2, \ldots, k_n$ is obtained by concatenating the interpolated features from the corresponding layers, that is, $\Phi_u(\mathbf{x}) = \Phi_u^{(k_1)}(\mathbf{x}) \oplus \Phi_u^{(k_2)}(\mathbf{x}) \oplus \cdots \oplus \Phi_u^{(k_n)}(\mathbf{x})$.

Our approach. Given a large unlabeled dataset, we first train representations using the instance-discriminative contrastive learning framework of MoCo [93]. A combination of geometric and photometric transformations are applied to generate pairs $(\mathbf{x}, \mathbf{x}')$. We then extract single layer or hypercolumn representations from the trained network to represent landmarks (Fig. 3.1c). Subsequently, we incorporate spatial contrastive learning to reduce dimensionality and induce spatial diversity by training a linear projector over the frozen landmark representation. Let $w \in \mathbb{R}^{C \times d}$, where $d \ll C$, used to project the landmark representation as $\Phi'_u(x) = w^T \Phi_u(x)$. The goal that the projected embeddings are spatially distinct within the same image,

$$\forall u, v \in \Omega : u \neq v \Leftrightarrow \Phi'_u(\mathbf{x}) \neq \Phi'_v(\mathbf{x}), \tag{3.6}$$

[1]Due to skip-connections, we cannot decompose the encoding over layers, but can across blocks.

is obtained by optimizing objective in Eqn. 3.3 with $\mathbf{x}' = \mathbf{x}$.

Discussion. Note that since the linear projection is location-wise, spatial equivariance is preserved but intra-image contrast is improved. The projected embeddings are equally effective as the hypercolumn representations for landmark regression, but are significantly better for landmark matching (Tab. 3.1). The intuition is that the hypercolumn features contain sufficient information about landmarks, but the projection step makes them spatially distinct which is more suitable for matching. Novotny *et al.* [161] proposed a similar approach to extract compact representations for cross-instance semantic matching from a network pre-trained with class labels. In comparison, we only use unsupervised representations. The idea of spatially contrastive learning has also been shown to be effective for learning scene-level representations [175].

Figure 3.2: Landmark matching with cosine distance using 3840-D hypercolumn and 256-D features projected from hypercolumn. Failure cases of using hypercolumns include (Left) mismatching between two eyes and (Middle) lack of robustness to the large viewpoint or (Right) appearance changes across different identities. The proposed feature projection method alleviates these issues.

3.1.5 Experiments

3.1.5.1 Benchmarks and implementation details

Human faces. We first compare the proposed model with prior art on the existing human face landmark detection benchmarks. Following DVE [229], we train our model on aligned CelebA dataset [136] and evaluate on MAFL [287], AFLW [115], and 300W [185]. The overlapping images with MAFL are excluded from CelebA. MAFL comprises 19,000 training images and 1000 test images with annotations on

5 face landmarks. Two versions of AFLW are used: AFLW_M which contains 10,122 training images and 2995 testing images, which are crops from MTFL [286]; AFLW_R which contains tighter crops of face images with 10,122 for training and 2991 for testing. 300W provides 68 annotated face landmarks with 3148 training images and 689 test images. We apply the same image pre-processing procedures as in DVE, the current state-of-the-art, for a direct comparison. We also train our model on the unaligned raw CelebA dataset to evaluate the efficiency of representation learning on in-the-wild unlabeled images.

Birds. We collect a challenging dataset of birds where objects appear in clutter and occlusion and exhibit wider pose variation. We randomly select 100K images of birds from the iNaturalist 2017 dataset [240] under the "Aves" class to train unsupervised representations. For the performance in the few-shot setting, we collect a subset of the CUB dataset [242] containing 35 species of *Passeroidea*[2] super-family, each annotated with 15 landmarks. We sample at most 60 images per class which result in 1241 images as our training set, 382 as the validation set, and 383 as the test set.

Evaluation. We use landmark matching and detection as the end tasks for evaluation. In landmark matching, following DVE [229], we generate 1000 pairs of images from the MAFL test set as the benchmark, among which 500 are pairs of the same identity obtained by warping images with thin-plate spline (TPS) deformation, and others are pairs of different identities. Each pair of images consists of a reference image with landmark annotations and a target image. We use the nearest neighbor matching with cosine distance between pixel representations for landmark matching, and report the mean pixel error between the predicted landmarks and the ground-truth landmarks.

[2]This is the biggest Aves taxa in iNaturalist.

Method	Dim.	Aligned		In-the-wild	
		Same	Diff.	Same	Diff.
DVE	64	0.92	2.38	1.27	3.52
Ours	3840	**0.73**	6.16	**0.78**	5.58
Ours + proj.	256	**0.71**	**2.06**	**0.96**	**3.03**
Ours + proj.	128	**0.82**	**2.19**	**0.98**	**3.05**
Ours + proj.	64	**0.92**	2.62	**0.99**	**3.06**

Table 3.1: Landmark matching results. We report the mean pixel error between the predicted landmarks and the ground-truth across 1000 pairs of images from MAFL (*lower is better*). The test set consists of 500 same-identity and 500 different-identity pairs. We compare DVE [229] with Hourglass net and our models with ResNet50 trained from aligned or in-the-wild CelebA dataset. We also evaluate the effect of feature projection (+proj.) with different output dimensions. Our results better than DVE's [229] are marked in bold.

In the landmark regression task, following [229, 230], we train a linear regressor to map the representations to landmark annotations while keeping the representations frozen. The landmark regressor is a linear regressor per target landmark. Each regressor consists of K filters of size $1\times1\times C$ on top of a C-dimensional representation to generate K intermediate heatmaps, which are then converted to spatial coordinates by `soft-argmax` operation. These K coordinates are finally converted to the target landmark by a linear layer. We use $K = 50$ to keep the evaluation consistent with prior works [229, 230], but we find that this hyperparameter is not critical (see Sec. 3.1.5.4). We report errors in the percentage of inter-ocular distance on face benchmarks and the percentage of correct keypoints (PCK) on CUB. A prediction is considered correct according to the PCK metric if its distance to the groundtruth is within 5% of the longer side of the image. The occluded landmarks are ignored during evaluation.

Implementation details. We use MoCo [93] to train our models on CelebA or iNat Aves for 800 epochs with a batch size of 256 and a dictionary size of 4096. ResNet18 or ResNet50 [96] are used as our backbones. We extract hypercolumns [88] per pixel by stacking activations from the second (`conv2_x`) to the last convolutional block

(`conv5_x`). We resize the feature maps from the selected convolutional blocks to the same spatial size as DVE [229] (*i.e.*, 48×48). We also follow DVE (with Hourglass network) to resize the input image to 136×136 then center-crop the image to 96×96 for face datasets. Images are resized to 96×96 without any cropping on the bird dataset. For a comparison with DVE on the CUB dataset we used their publicly available implementation.

3.1.5.2 Landmark matching

Quantitative results. Tab. 3.1 compares the proposed method with DVE [229] quantitatively. We train DVE and our models on both aligned and in-the-wild unaligned versions of the CelebA dataset, and report the mean pixel error on aligned face images from MAFL. Our hypercolumn representation has high performance in same-identity matching but is not robust to cross-identity variations. However, the proposed feature projection makes the hypercolumn more suitable for landmark matching. We experiment with different feature dimensions after projection and find that our method with 128 or higher dimensional features achieves the state-of-art. DVE outperforms ours with 64-D features when the representations are learned on the aligned CelebA dataset. This is because the architecture of the Hourglass network and the joint training of the backbone and feature extractor enables DVE to learn a more compact representation than our method. However, to lift the feature dimension from 64 to 256, DVE requires re-training the entire model while we only need to re-train a linear feature projector. Moreover, when the representation is learned from the in-the-wild CelebA dataset, our model *outperforms DVE by a large margin.* This suggests our representation is more invariant to nuisance factors than that of DVE. We also observe that our method with smaller networks (*e.g.*, ResNet18 [96]) with 128-D projected features outperforms DVE, and both DVE and our methods outperform representations from ImageNet pretrained networks.

Method	# Params. Millions	Unsuper.	MAFL	AFLW$_M$	AFLW$_R$	300W	CUB
			Inter-ocular Distance (%) ↓				PCK ↑
TCDCN [287]	–	×	7.95	7.65	–	5.54	–
RAR [262]	–	×	–	7.23	–	4.94	–
MTCNN [286]	–	×	5.39	6.90	–	–	–
Wing Loss [66]	–	×	–	–	–	4.04	–
Generative modeling based							
Structural Repr. [284]	–	✓	3.15	–	6.58	–	–
FAb-Net [249]	–	✓	3.44	–	–	5.71	–
Deforming AE [202]	–	✓	5.45	–	–	–	–
ImGen. [105]	–	✓	2.54	–	6.31	–	–
ImGen.++ [106]	–	✓	–	–	–	5.12	–
Equivariance based							
Sparse [231]	–	✓	6.67	10.53	–	7.97	–
Dense 3D [230]	–	✓	4.02	10.99	10.14	8.23	–
DVE SmallNet [229]	0.35	✓	3.42	8.60	7.79	5.75	–
DVE Hourglass [229]	12.61	✓	2.86	7.53	6.54	4.65	61.91
Invariance based							
Ours (ResNet18)	11.24	✓	**2.57**	8.59	7.38	5.78	**62.24**
Ours (ResNet18 + proj.)	11.24	✓	**2.71**	**7.23**	**6.30**	5.20	58.49
Ours (ResNet50)	23.77	✓	**2.44**	**6.99**	**6.27**	5.22	**68.63**
Ours (ResNet50 + proj.)	23.77	✓	**2.64**	**7.17**	**6.14**	4.99	**62.55**

Table 3.2: Results on landmark detection. Comparison on face benchmarks, including MAFL, AFLW$_M$, AFLW$_R$, and 300W, and CUB dataset. We report the error in the percentage of inter-ocular distance on the human face dataset (*lower is better*), and the percentage of correct keypoints (PCK) on the CUB dataset (*higher is better*). We project the hypercolumn (*i.e.*, + proj.) to 256-D features on the face and 512-D on the bird dataset. Our results better than DVE's [229] are marked in bold.

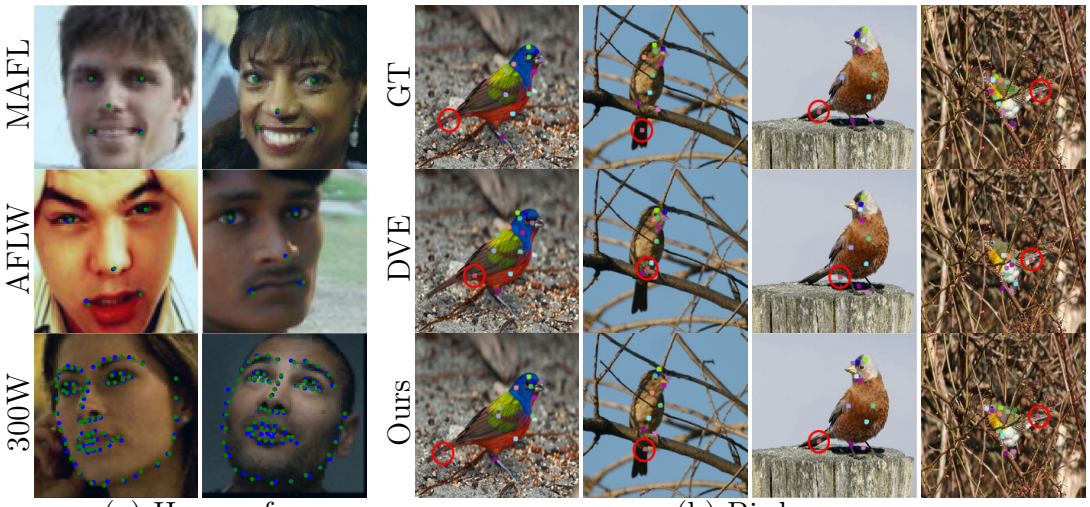
Qualitative results. Fig. 3.2 presents the qualitative results of landmark matching. Our method with hypercolumn for matching is not robust to viewpoint and appearance changes and frequently mismatches the left and right eyes. Incorporating the proposed feature projection adds diversity effectively and solves these issues.

3.1.5.3 Landmark detection

Quantitative results. Tab. 3.2 presents a quantitative evaluation of multiple benchmarks. On faces, our model with a ResNet50 achieves state-of-the-art results on all benchmarks except for 300W. On iNat Aves → CUB, out approach outperforms prior state-of-the-art [229] by a large margin, suggesting improved invariance to nuisance

factors. Incorporating the feature projection results in small performance degradation in some cases but remains the state-of-art. Our method with ResNet18 is comparable with DVE and benefits from using a deeper network.

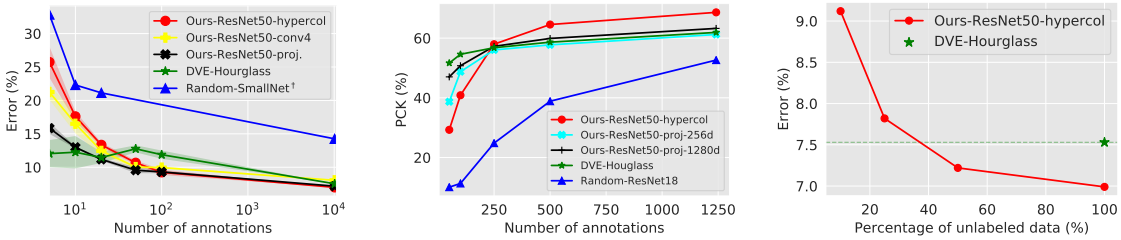
Qualitative results. Fig. 3.3 shows qualitative results of landmark regression on human faces and birds. We notice that both DVE and our model with hypercolumn representations are able to localize the foreground object accurately. However, our model localizes many keypoints better (*e.g.*, on the tails of the birds) and is more robust to the background clutter (*e.g.*, the last column of Fig. 3.3b).



(a) Human face (b) Birds

Figure 3.3: Detected landmarks (a) on faces (*blue:* predictions, *green:* ground truth) and (b) on CUB. Notice that our method localizes the tails of birds (circled) much better. *Zoom in for details.*

Limited annotations. Fig. 3.4a and 3.4b compare our model with DVE [229] using a limited number of annotations on AFLW$_M$ and CUB dataset respectively. Without feature projection, our performance is better as soon as a few training examples are available (*e.g.*, 50 on AFLW$_M$ and 250 on CUB). This can be attributed to the higher dimensional embedding of the hypercolumn representation. The scheme can be improved by using a single-layer representation as shown in the yellow line. Our feature projection further improves the performance in the low-data regime as shown in the black line. Interestingly, this improvement is not solely due to the dimension



(a) Limited anno. on AFLW$_M$(b) Limited anno. on CUB(c) Unlabeled CelebA images

Figure 3.4: The effect of dataset size. (a) A comparison of our model with DVE [229] by varying the number of annotations for landmark regression on AFLW$_M$ dataset. `Random-SmallNet`†: is a randomly initialized "small network" taken from [229]. `Ours-ResNet50`: is based on hypercolumn, or its compact representations, or fourth-layer features trained using contrastive learning. (b) Similar results on CUB dataset. `Random-ResNet18`: is trained from scratch on the CUB dataset. (c) Results of landmark regression on AFLW$_M$ using different numbers of *unlabeled* images from CelebA for training.

reduction: increasing the dimension of the projected feature from 256 to 1280 improves the performance across different dataset sizes on CUB (see Fig. 3.4b). Note that all unsupervised learning models (including DVE and our model) outperform the randomly initialized baseline on both the human face and bird datasets.

Dataset	Single layer					Hypercolumn			
	#1 (64)	#2 (256)	#3 (512)	#4 (1024)	#5 (2048)	#4 - #5 (3072)	#3 - #5 (3584)	#2 - #5 (3840)	#1 - #5 (3904)
MAFL	5.77	4.58	3.03	**2.73**	3.66	2.73	2.65	**2.44**	2.51
AFLW$_M$	24.20	21.34	11.95	**8.83**	11.55	8.14	8.31	**6.99**	7.40
AFLW$_R$	16.27	14.15	9.66	**7.37**	8.83	6.95	**6.24**	6.27	6.34
300W	16.45	13.08	7.66	**6.01**	7.70	5.68	5.28	5.22	**5.21**

Table 3.3: Landmark detection using single layer and hypercolumn representations. The error is reported in the percentage of inter-ocular distance using linear regression over individual layers (left) and combinations (right), with a ResNet50. The embedding dimension for each is shown in parentheses. Layer #4 performs the best across datasets, while hypercolumns offer an improvement.

Limited unlabeled data. Fig. 3.4c shows that our model with hypercolumn representation matches the performance of DVE on AFLW$_M$ using only 40% of the images

on the CelebA dataset. This suggests that invariances are acquired more efficiently in our framework.

3.1.5.4 Ablation studies and discussions

Hypercolumns. Tab. 3.3 compares the performance of using individual layer and hypercolumn representations. The activations from the fourth convolutional block consistently outperform those from the other layers. For an input of size 96×96, the spatial dimension of the representation is 48× 48 at Layer #1 and 3×3 at Layer #5, reducing by a factor of two at each successive layer. Thus, while the representation loses geometric equivariance with depth, contrastive learning encourages invariance, resulting in Layer #4 with the optimal trade-off for this task. While the best layer can be selected with some labeled validation data, the hypercolumn representation provides further benefits everywhere except the very small data regime (Tab. 3.3 and Fig. 3.4a).

Dimensionality and linear regressor. In Tab. 3.4, we reduce the size of the landmark regressor to evaluate its effect on the landmark regression performance. We chose 50 intermediate landmarks to keep the evaluation consistent with DVE. However, the choice is not critical as seen by the performance of a smaller linear regressor. There is a small drop in performance, while it remains comparable to DVE. The proposed feature projection with equivariant learning is more effective than non-negative matrix factorization (NMF), a classical dimension reduction method.

Effectiveness of unsupervised learning. Tab. 3.5 compares representations using the linear evaluation setting for randomly initialized, ImageNet pretrained, and contrastively learned networks using a hypercolumn representation. Contrastive learning provides significant improvements over ImageNet pretrained models, which is less surprising since the domain of ImageNet images is quite different from faces. Interestingly, random networks have competitive performances with respect to some prior

Method	C	K	#P	MAFL	AFLW$_M$	AFLW$_R$	300W
DVE	64	50	17	2.86	7.53	6.54	4.65
Ours	3840	50	961	**2.44**	**6.99**	**6.27**	5.22
Ours	3840	10	192	**2.40**	**7.27**	**6.30**	5.40
Ours+proj.	256	50	65	**2.64**	**7.17**	**6.14**	4.99
Ours+proj.	256	10	13	**2.67**	**7.24**	**6.23**	5.07
Ours+proj.	64	50	17	**2.77**	**7.21**	**6.22**	5.19
Ours+NMF	64	50	17	**2.80**	7.60	6.69	5.62

Table 3.4: The effect of landmark regressor on landmark regression. We vary the number of parameters (#P in thousands) in the landmark regressor by changing the number of intermediate landmarks (K) and feature dimensions (C). We compare the proposed feature projection (*i.e.*, +proj.) with non-negative matrix factorization (NMF) for dimension reduction. Our results better than DVE's [229] are marked in **bold**.

work in Tab. 3.2. For example, [230] achieve 4.02% on MAFL, while a randomly initialized ResNet18 with hypercolumns achieves 4.00%.

Network	Supervision	MAFL	AFLW$_M$	AFLW$_R$	300W
Res. 18	Random	4.00	14.20	10.11	9.88
	ImageNet	2.85	8.76	7.03	6.66
	Contrastive	2.57	8.59	7.38	5.78
Res. 50	Random	4.72	16.74	11.23	11.70
	ImageNet	2.98	8.88	7.34	6.88
	Contrastive	**2.44**	**6.99**	**6.27**	**5.22**

Table 3.5: Effectiveness of unsupervised learning. Error using randomly initialized, ImageNet pretrained, and contrastively trained ResNet50 for landmark detection. Frozen hypercolumn representations with linear regression were used for all methods.

Are the learned representations semantically meaningful? We found that parts can be reliably distilled from the learned representation using non-negative matrix factorization (NMF) (see [47] for another application of NMF for visualizing semantic parts from deep network activations). Fig. 3.5 shows two such components and a "map" of several components, which are indicative of parts (left) and are robust

Figure 3.5: Semantic parts distillation. The object parts distilled from our representation using NMF are semantically meaningful and consistent across different instances (left). The parts are also robust to geometric transformations (right).

to image transformations (right). Additionally, Fig. 3.2 shows that the correspondence obtained using nearest neighbor matching are semantically meaningful.

Commonalities and differences. Equivariance is necessary but not sufficient for an effective landmark representation. It also needs to be distinctive or invariant to nuisance factors. This is enforced in the equivariance objective (Eqn. 3.3) as a contrastive term over locations within the same image, as the loss is minimized when $p(v|u; \Phi, \mathbf{x}, \mathbf{x}')$ is maximized at $v = gu$. This encourages intra-image invariance, unlike the objective of contrastive learning (Eqn. 3.5) which encourages inter-image invariance. However, a single image may contain enough variety to guarantee some invariance. This is supported by its empirical performance and recent work showing that representation learning is possible even from a single image [274]. However, our experiments suggest that inter-image invariance can be more effective on datasets with greater clutter, occlusion, and pose variations.

Is there any advantage of one approach over the other? Our experiments show that for a deep network of the same size, invariant representation learning can be just as effective (Tab. 3.2). However, invariant learning is conceptually simpler and scales better than equivariance approaches, as the latter maintains high-resolution feature maps across the hierarchy. Using a deeper network (*e.g.*, ResNet50 vs. ResNet18)

gives consistent improvements, outperforming DVE [229] on four out of five datasets, as shown in Tab. 3.2. A drawback of our approach is that the hypercolumn representation is not directly interpretable or compact, which results in lower performance in the extreme few-shot case. However, as seen in Fig. 3.4a, the advantage disappears with as few as 50 training examples on the AFLW benchmark. This problem can be effectively alleviated by learning a compact representation using equivariant learning which further reduces the number of required training examples to 20. Invariant learning is also more data-efficient and can achieve the same performance with half the unlabeled examples, as seen in Fig. 3.4c.

3.1.6 Conclusion and subsequent works

We show that intermediate layer representations of a deep network trained using instance-discriminative contrastive learning outperform landmark representation learning approaches that are based on unsupervised equivariant learning alone. We also show that equivariant learning approaches can be viewed through the lens of (spatial) contrastive learning, resulting in weaker generalization than inter-image invariances for landmark recognition tasks. However, these two forms of contrastive learning are complementary and we use the latter to learn a compact representation that is better suited for landmark matching tasks. We illustrate our results on existing benchmarks and a new challenging one where there is a larger variation in pose and viewpoint, where the improvements using our approach are more pronounced.

Following the publication of our research in 2021, there have been several noteworthy advancements in the field. Our work shares a broad relation with studies exploring emergent properties of training deep neural networks without explicit human annotations, particularly those aimed at understanding object structures. Herein, we provide a brief overview of these recent developments.

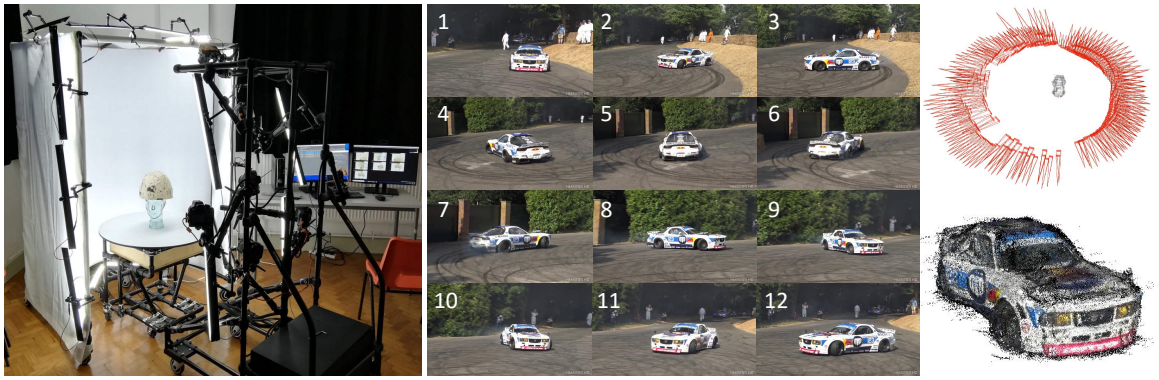
Xu *et al.* [268] demonstrated that object-level and part-level semantic correspondence emerges when the ResNet [96] is trained with the image-level contrastive learning, resonating our observations. In parallel, Zhang *et al.* [285] showed that the intermediate features from a pretrained StyleGAN [111] are highly predictive of semantic parts and developed a few-shot part segmentation framework.

Naturally, this brings up the question of which unsupervised pretraining strategy, StyleGAN or contrastive learning, is more effective and efficient in understanding object structure. We sought to answer this question in collaboration with colleagues, conducting extensive evaluations which revealed the contrastive learning approach to be superior to StyleGAN-based methods on standard few-shot part segmentation benchmarks [186]. More recently, the diffusion model [101, 207] has greatly advanced the field of generative modeling. Baranchuk *et al.* [11] have demonstrated that internal representations from pretrained diffusion models [101] outperform previous state-of-the-art methods, including GANs [111] and self-supervised learning models [92]. As of July 2023, DINO [26], a self-supervised learning strategy, stands as the leading method for understanding semantic parts, as demonstrated by Amir *et al.* [4].

In addition to the aforementioned pretraining strategies, another critical component of our framework is the feature projection (Eqn. 3.6), which significantly improves landmark matching performance over raw hypercolumn features (Tab. 3.1). Most recently, Aygun *et al.* [9] conducted extensive evaluations of various feature projection algorithms (including ours) across multiple benchmarks, further verifying the efficacy of our method and introducing an improved approach.

3.2 Learning 3D pose estimators from unlabeled videos

In this section, we introduce a technique for learning single-view 3D object pose estimation models by utilizing a new source of data — in-the-wild videos where objects turn. Such videos are prevalent in practice (*e.g.*, cars in roundabouts, airplanes



(a) Classic turntable (b) Accidental turntable

Figure 3.6: Classic turntable vs. accidental turntable. (a) Classic turntables rotate and scan objects in a controlled environment to estimate their 3D pose and shape. (b) A turning object in a video leads to an accidental turntable. Structure-from-motion, coupled with object detection [95] and feature matching [191], provides surprisingly accurate relative 3D pose estimation (top) and 3D reconstruction (bottom) — the red pyramids indicate the estimated relative poses of video frames. We utilize a collection of such videos to train and evaluate models for single-frame 3D pose estimation in realistic settings. See more accidental turntables here: `https://www.youtube.com/watch?v=8rFNRri8-TI`

near runways) and easy to collect. We show that classical structure-from-motion algorithms, coupled with the recent advances in instance detection and feature matching, provide surprisingly accurate relative 3D pose estimation on such videos. We propose a multi-stage training scheme that first learns a canonical pose across a collection of videos and then supervises a model for single-view pose estimation. The proposed technique achieves competitive performance with respect to the existing state-of-the-art on standard benchmarks for 3D pose estimation without requiring any pose labels during training. We also contribute an Accidental Turntables Dataset, containing a challenging set of 41,212 images of cars in cluttered backgrounds, motion blur, and illumination changes that serve as a benchmark for 3D pose estimation.

3.2.1 Overview

Mechanical devices that precisely change an object's pose are widely utilized when performing high-precision 3D scanning. They allow a particular object to have its

pose modified in a controlled manner while capturing its appearance through a variety of image sensors. One of the simplest devices of this kind is a *turntable* – a rotating platform that slowly changes the pose of an object through an electric motor (Fig. 3.6a). Unfortunately, despite its simplicity, *turntables* are not very practical. They need to be as large as the object at hand, *e.g.*, setting up turntables for cars or airplanes would require a lot of work.

Fortunately, we don't need to place those objects in actual turntables. Many are already performing similar motions on their own (Fig. 3.6b) — cars moving along roundabouts, airplanes landing and parking, ships maneuvering across canals, and so on. In the real world, video recordings of objects performing these types of motions depict them in uncontrolled environments; *i.e.*, cluttered background, occluders, changes in illuminations, motion blur, unpredictable pose changes, and many other nuisance factors. Thanks to many recent advances in computer vision, we show that we are able to bypass many of those nuisance factors and apply Structure from Motion (SfM) to reliably and precisely recover relative pose estimation from videos of real objects (Fig. 3.6b). We call these types of videos **Accidental Turntables** – objects presenting motion patterns that allow us to observe them from (almost) all possible angles. We demonstrate that these videos, after suitable automatic pre-processing, are an excellent source of supervision for pose estimation models and, perhaps more importantly, can be mined from the internet, enabling the creation of bigger and more diverse datasets.

However, using the supervision from SfM does not allow us to directly perform pose estimation with respect to a canonical object frame. To this end, we propose to learn a *relative* pose estimation model and show that its training leads to the emergence of a canonical object pose. In the second stage, we propose a calibration and training procedure that allows pose estimation in a canonical frame. We show that models trained in this fashion *only* using our newly collected dataset from *real*

videos significantly outperform other models trained on SfM and perform on par with existing unsupervised approaches on standard benchmarks, *e.g.*, the Freiburg and ImageNet cars datasets.

We summarize our contributions as follows. 1) a procedure for automatically processing accidental turntable videos and annotating its frames with relative pose transformations; 2) a multi-stage training scheme that allows training accurate pose estimation models with respect to arbitrary canonical frames; and 3) a new dataset with 41,212 real images of cars from turntable videos with their corresponding pose annotation.

3.2.2 Related works

Datasets for 3D pose estimation. A number of datasets provide 3D pose annotations for objects in the wild [3, 70, 208, 214, 259, 260] or in controlled environments [65, 102, 243, 248, 261]. These datasets have been widely used for training supervised pose estimation models [80, 125, 141, 236]. However, manually annotating 3D poses is very tedious and thus not scalable. Unsupervised pose estimation models [143, 155, 162, 197] learn to predict 3D pose without any human annotations. Videos [166, 197] that capture multiple views of objects have been the main source of training data in prior works [143, 162, 197]. However, to acquire such videos, a person needs to hold a camera and slowly move around a *static* object. This is a time-consuming procedure, especially for large-size objects (*e.g.*, cars, and airplanes), and has limited the size of existing video datasets. For example, the Freiburg Cars dataset [197] consists of 52 car videos, and EPFL car dataset [166] only provides 20 cars. Such limited data may further constrain the performance of prior methods.

Supervised 3D pose estimation. With groundtruth 3D pose annotations, supervised pose estimation works have been focusing on developing novel representations of 3D pose [125, 154, 290], learning objectives [125, 236, 265, 266], or network ar-

chitectures [61, 62]. The difficulty in annotating 3D poses results in the scarcity of pose annotations. This issue is partially relieved by augmenting the existing datasets with synthetic data [210]. The integration of pose estimation and object detection has been explored in the task of 3D object detection [57, 70].

Unsupervised 3D pose estimation. Unsupervised pose estimation models learn 3D object pose without any human annotations. Prior works are either based on analysis-by-synthesis [143, 155] or SfM [162, 197]. The analysis-by-synthesis frameworks train a pose estimation model by reconstructing the input images in a pose-aware manner. The SfM-based methods start by estimating the pose labels with SfM on videos that capture 360° views of static objects. However, SfM only provides relative pose estimations among video frames. The absolute pose estimations from SfM are not consistent across videos (*i.e.*, objects in the same pose from two videos may have quite different absolute pose representations). To tackle this issue, Sedaghat *et al.* [197] calibrate the SfM pose estimations via aligning 3D reconstructions of objects; Novotny *et al.* [162] train a model to estimate the relative pose and observe that canonical poses emerge in the models trained in this manner.

3.2.3 Accidental Turntables dataset

In this section, we provide the details of our data collection and the generation of 3D pose annotations with SfM algorithms on our dataset. We name the collected video dataset as **Accidental Turntables dataset**, highlighting its connections to classic turntables (Fig. 3.6).

Data source. The main criterion of our data collection is that the object turns in the video. Such videos are abundant on the Internet and quite easy to acquire. In this work, we focus on the car category which is one of the most common moving objects in the wild (at least in America). We leave the extension to other categories (*e.g.*, airplanes and boats) in our future work but include some examples of the

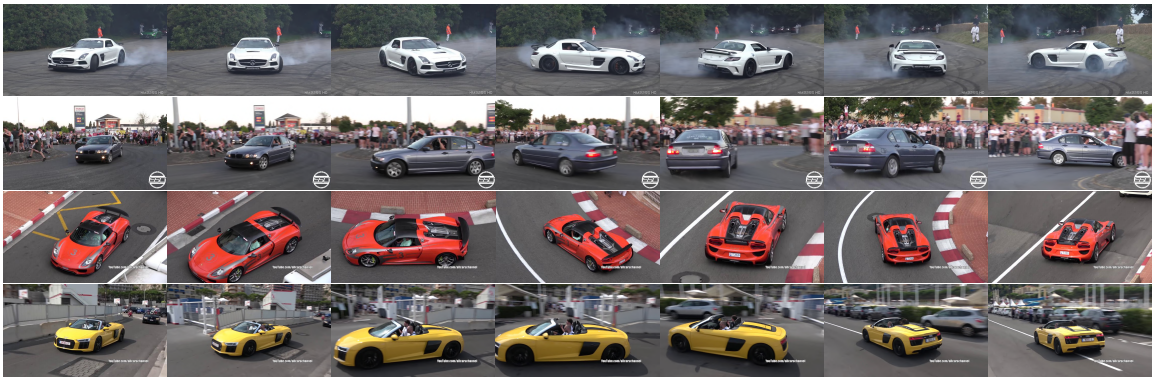


Figure 3.7: Samples from Accidental Turntables dataset. Accident turntables are prevalent in practice. For instance, a car donuts (1st row), a car moves along a roundabout (2nd and 3rd row), or a car does not turn but passes by a camera (4th row). All car instances exhibit at least 180° azimuth changes relative to the camera.

reconstructions in Sec.3.2.5.3. We collect 313 car video clips from YouTube containing a total of 141,784 frames. Each video consists of a single moving car instance that exhibits multiple views in motion. Fig. 3.7 provides video samples from our dataset.

Challenges. Even though our dataset consists of a large number of car videos serving as a new source of training data for machine learning models, in-the-wild videos pose technical challenges for the automatic extraction of 3D poses using SfM. For example, to exploit the classical SfM algorithms to estimate the object pose, object segmentation is required to remove the background; Motion blur and texture-free object surfaces necessitate robust interest points detection; Discriminative feature description and robust feature matching are needed to avoid the ambiguity of pose estimation on symmetric objects (*e.g.*, cars).

Pose estimation with SfM. To tackle the above-mentioned challenges, we use the MaskRCNN [95] pretrained on MS-COCO dataset [127] to remove the background clutter. We find that the MaskRCNN provides highly accurate object detection and segmentation on in-the-wild car videos. We use SfM algorithms implemented by COLMAP [194, 195] with SuperPoint [56] as the feature extractor and SuperGLUE [191] as the feature matcher to estimate the object pose on cropped object

images. We sequentially match the next 10 frames per video frame, instead of exhaustively matching every pair of frames in a video. Sequential matching reduces the ambiguity in matching repeated patterns (*e.g.*, left and right wheels of a car). SfM, coupled with MaskRCNN, SuperPoint, and SuperGLUE, provides surprisingly accurate pose estimation, in comparison with classical SIFT [138] and nearest neighbor matching. We provide a detailed study on the effect of feature extraction and matching on SfM in Sec.3.2.5.3.

Statistics. Our dataset consists of 313 car videos with 141,784 frames in total. SfM automatically samples frames with sufficient large relative pose change and reliable feature matching. Adjacent frames in a video usually have tiny differences in the pose. Thus, most of the frames are filtered out by SfM. We end up collecting 41,212 frames with SfM pose estimations. Our dataset covers cars with diverse shapes, colors, textures, and poses (see examples in Fig. 3.7).

3.2.4 Approach

This section introduces our framework for learning 3D object pose from the proposed Accidental Turntables dataset. Fig. 3.8 illustrates an overview of the proposed framework. SfM estimates the relative pose of objects with respect to the object in the first frame per video, followed by optimizing the pose parameters with the bundle adjustment. However, the object pose in the first frame may vary dramatically across videos. It is thus meaningless to train a model directly on the absolute pose labels from SfM. Instead, we start by training a model to estimate the *relative* pose of frame pairs (Fig. 3.8 left). We observe that a canonical pose emerges in our pose estimation model train in this way (see Sec.3.2.5.3). This provides us a tool to calibrate the pose estimation from SfM to a canonical frame (Fig. 3.8 middle). In the second stage, we train a pose estimation model directly on the calibrated *absolute* pose annotations similar to standard supervised learning methods [210, 236, 265] (Fig. 3.8 right). We

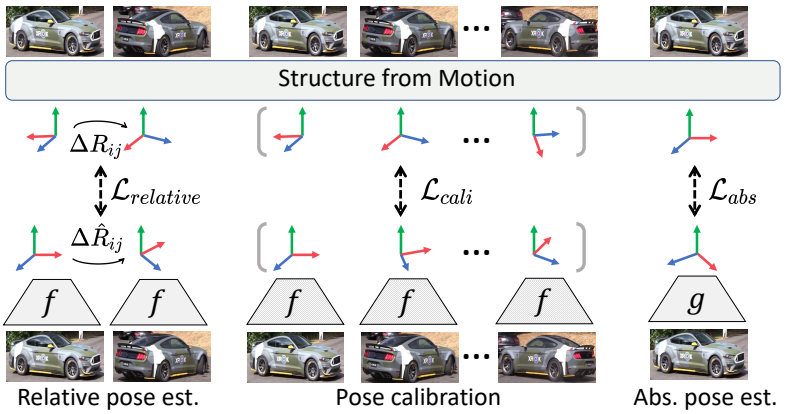


Figure 3.8: Approach overview. Left: a pose estimation model $f(x)$ is trained to predict the *relative* pose of image pairs (denoted by ΔR_{ij}). Middle: the emergence of the canonical pose in $f(x)$ enables us to calibrate the pose estimations from SfM to a uniform frame. The model $f(x)$ is frozen in the pose calibration step. Right: after the pose calibration, a pose estimation model $g(x)$ is trained on the *absolute* pose annotations.

denote our model trained in the first stage as $f(x)$ and the model in the second stage as $g(x)$, where x is the input image. Our Accidental Turntables dataset is denoted by $\{(x_i, R_i)\}$, where $R \in \mathbf{SO}(3)$ is the SfM pose estimation.

Relative pose estimation. In this stage, we train a single-view pose estimation network $f(x)$ to predict the relative pose between pairs of video frames. The loss function is defined as

$$\mathcal{L}_{\text{relative}} = \sum_{(i,j)}^{N} \text{dist}(R_i R_j^T, \hat{R}_i \hat{R}_j^T) \quad \text{with} \quad \hat{R}_i = f(x_i) \tag{3.7}$$

where $\text{dist}(\cdot, \cdot)$ is a distance function between two rotation matrices (*e.g.*, L_2 or geodesic distance). $\hat{R}_i$ is a 3×3 rotation matrix predicted from the model $f(x_i)$ on the input x_i. The frame pair x_i and x_j are sampled from the same video. N is the total number of frame pairs sampled from our video dataset. $\Delta R_{ij} = R_i R_j^T$ is the relative rotation matrix that transforms the pose of the frame x_j to x_i. We use the 6D continuous rotation representation [290] as the intermediate output of our model

$f(x)$, from which the 3×3 rotation matrices $\hat{R}$ are recovered by the Gram-Schmidt orthogonalization [290]. Our first training stage is similar to the learning strategy proposed by Novotny *et al.* [162]. Differently, we only use the model $f(x)$ trained in this stage as a tool to calibrate the SfM pose annotations (Sec. 3.2.4). Moreover, we demonstrate that the model $g(x)$ trained in our second stage significantly outperforms the stage-one model $f(x)$ as well as Novotny *et al.* [162].

Pose calibration. The pose predictor $f(x)$ trained in the first stage provides us a tool to calibrate the pose annotations from SfM into a uniform pose frame, thanks to the emergence of canonical pose (see Sec.3.2.5.3 for more details). If the pretrained $f(x)$ provides perfectly accurate pose estimation per input x, there exists a global rotation ΔR for each video that aligns our pose annotations $\{R_i\}$ to the pose predictions $\{\hat{R}_i\}$:

$$\hat{R}_i = \Delta R R_i \quad \forall i \in 1, \ldots, K \tag{3.8}$$

Where K is the number of frames in the target video, however, the pose predictions $\{\hat{R}_i\}$ are inaccurate in practice due to the limited performance of the pretrained pose predictor $f(x)$. We thus target at a rotation matrix ΔR^* that aligns $\{R_i\}$ and $\{\hat{R}_i\}$ with minimal calibration error. We define the calibration error as,

$$\mathcal{L}^*_{cali} = \frac{1}{K} \sum_{i}^{K} \text{dist}(\hat{R}_i, \Delta R^* R_i) \tag{3.9}$$

where $\text{dist}(\cdot, \cdot)$ is a distance function between two rotation matrices. We adopt the geodesic distance $\| \log R^T \hat{R} \|_{\mathcal{F}} / \sqrt{2}$ in our implementation. The pose calibration is then formulated as an optimization problem:

$$\min_{\Delta R} \quad \mathcal{L}_{\text{cali}}(\hat{R}, \Delta R R) \tag{3.10}$$

$$\text{s.t.} \quad \Delta R \in \mathbf{SO}(3) \tag{3.11}$$

This problem can be solved by the classical Procrustes analysis [78]. In practice, we find that a simple search-based optimization method works reliably. Concretely, the optimal global rotation ΔR^* is searched from the set $\{\Delta R_j : \Delta R_j = \hat{R}_j R_j^T\}$. Moreover, the calibration error L^*_{cali} is closely related to the noise level of the calibrated pose annotations. Large calibration error typically means the failure of calibration and a higher level of noise in the calibrated pose annotations (see Sec.3.2.5.3 for our empirical studies). Therefore, the calibration error L^*_{cali} may serve as a heuristic to filter out noisy pose labels.

Absolute pose estimation. We now could apply any supervised learning methods for pose estimation on our calibrated dataset $\{(x_i, R_i^{\text{cali}})\}$. In this work, we adopt the framework proposed by Xiao *et al.* [265, 266] to train our pose estimator. Concretely, we use three Euler angles as our pose representation, including azimuth $\alpha \in [-\pi, \pi]$, elevation $\beta \in [-\pi/2, \pi/2]$, and roll $\gamma \in [-\pi, \pi]$. The Euler angles are decomposed from the rotation matrices R^{cali} and divided into Z_θ disjoint angular bins with bin size $B_\theta = \pi/12$. The model is trained to predict the bin indices $y_\theta \in \{1, \ldots, Z_\theta\}$ via a classification loss and within-bin offsets δ_θ via a regression loss:

$$\mathcal{L}_{\text{abs}} = \sum_{\theta \in \alpha, \beta, \gamma} \mathcal{L}_{cls}(y_\theta, p_\theta) + \lambda \mathcal{L}_{\text{reg}}(\delta_\theta, \hat{\delta}_\theta) \tag{3.12}$$

where p_θ is the probability of the object pose in the bin y_θ; $\hat{\delta}_\theta \in [0, 1]$ is the predicted offsets within the bin y_θ; $(p_\theta, \hat{\delta}_\theta) = g(x)$ are both outputs of our pose estimation model $g(x)$. We use the cross-entropy loss as the classification loss $\mathcal{L}_{cls}$ and the smooth-L1 loss as the regression loss $\mathcal{L}_{\text{reg}}$; λ is the weight on the regression loss ($\lambda = 1$ by default).

At the inference time, the pose prediction $\hat{\theta}$ on the input x is obtained by combining the prediction of the bin classifier and the offsets within the predicted angular bin:

$$\hat{\theta} = (j + \hat{\delta}_{\theta,j})B_\theta \quad \text{with} \quad j = \underset{i}{\operatorname{argmax}}\, p_{\theta,i} \tag{3.13}$$

where $p_{\theta,i}$ is the probability of object pose in the i-th bin, and $\hat{\delta}_{\theta,j}$ is the predicted offsets within the i-th bin.

3.2.5 Experiments

3.2.5.1 Benchmark and implementation details

Implementation details. We use a standard ResNet50 network with three fully-connected layers as our pose estimation model. We initialize our model with ImageNet pretrained weights and fine-tune it during training. In the first training stage, we do not apply any data augmentation. In the second training stage, we use standard data augmentations including in-plane rotation and flipping. We conduct hyperparameter search and checkpoint selection on a validation set separate from our training and test set. The validation set consists of 338 non-truncated and non-occluded car images from PASCAL3D+ [260]. Similar to prior work [143, 155, 265, 266], we use a tightly cropped object image as the input to our pose estimation model. The input image is resized and padded to 224×224. We use the Adam optimizer [112] with a learning rate of 1E-4 and weight decay of 5E-4. In the second training stage, we train our model on videos with a calibration error $\mathcal{L}^*_{\text{cali}}$ (Eqn. 3.9) lower than 7°.

Benchmarks. We evaluate the performance of our model on the PASCAL3D+ dataset [260] which is a standard benchmark for 3D pose estimation. The test split in the PASCAL3D+ dataset consists of 308 non-occluded and non-truncated car images collected from the PASCAL VOC dataset [63]. More recently, Mariotti *et al.* [143] reports their results on the ImageNet validation set included in PASCAL3D+ which consists of 2712 test images of cars. To make a comparison with Mariotti *et al.* , we provide results on both test splits. Following prior works, we measure the prediction error using the standard geodesic distance $\Delta R = \| \log R_{gt}^T R_{pred} \|_{\mathcal{F}} / \sqrt{2}$ between the estimated rotation matrix R_{pred} and the groundtruth R_{gt}. We report the median

geodesic error (Med.) and the percentage of predictions with error less than $\pi/6$ (Acc.) relative to the groundtruth.

Pose calibration for evaluation. The pose predictions from our model align with human annotations up to a global rotation, due to the difference between the coordinate frame of our model and that of pose annotation tools adopted by the benchmarks. To evaluate our model on the benchmarks, similar to prior unsupervised learning methods [143, 155], we need to calibrate our pose estimations to the groundtruth annotations. Such pose calibration for evaluation is exactly the same as our pose calibration step described in Sec 3.2.4. Specifically, we estimate a global calibration matrix ΔR such that $\Delta R R_{pred}$ equals the human annotations R_{gt}. We formulate the pose calibration as an optimization problem and solve it via a simple search-based method (see more details in Sec 3.2.4). The calibration matrix ΔR is obtained by solving the optimization problem on 100 car images randomly sampled from the training set of PASCAL3D+.

3.2.5.2 Single-view 3D pose estimation

Quantitative results. Tab. 3.6 provides quantitative comparisons with prior unsupervised pose estimation works on PASCAL3D+ test set. Our method significantly outperforms the existing SfM-based methods [162, 197]. Similar to ours, these models are trained on video data with pose annotations from SfM. However, they rely on SfM with SIFT [138] and nearest neighbor (NN) matching, which fails to provide high-quality pose estimations (see more details in Sec.3.2.5.3). For this reason, prior SfM-based models collect videos by slowly moving a camera around *static* cars to avoid large motion blur. This tedious procedure limits the size of existing car video datasets. For example, the FreiburgCars dataset [197] consists of 52 car videos; the EPFL car dataset [166] provides only 20 car videos. In comparison, our video dataset (consisting of 313 videos) is easy to collect and prevalent on the Internet.

	Methods	Supervision	Trainset	Testset	Acc.(%) ↑	Med.(°) ↓
Super.	Tulsiani *et al.* [236]	Anno.	PASCAL3D+	VOC	89	9.1
	Mahendran *et al.* [141]	Anno.	PASCAL3D+	VOC	–	8.1
	Liao *et al.* [125]	Anno.	PASCAL3D+	VOC	93	5.2
	Grabner *et al.* [80]	Anno.	PASCAL3D+	VOC	94	5.1
Unsupervised	VPNet [197]	SfM	FreiburgCars	VOC	–	49.6
	VpDRNet [162]	SfM	FreiburgCars	VOC	–	29.6
	SSV [155]	AbS	CompCars	VOC	67	10.1
	Ours	SfM	FreiburgCars	VOC	72	15.7
	Ours	SfM	Acci.Turn.	VOC	75	15.8
	ViewNet* [143]	AbS	ShapeNet	ImageNet	88	5.6
	ViewNet* [143]	AbS	FreiburgCars	ImageNet	61	16.1
	Ours	SfM	FreiburgCars	ImageNet	84	15.0
	Ours	SfM	Acci.Turn.	ImageNet	86	14.8

Table 3.6: Pose estimation on PASCAL3D+ test sets. We make comparisons with supervised learning methods trained with human annotations (dubbed Anno.) and unsupervised pose estimation models based on Structure-from-Motion (dubbed SfM) or Analysis-by-Synthesis (dubbed AbS). *ViewNet ignores the in-plane rotation in the evaluation and reports the results on the ImageNet validation set.

SfM, coupled with the recent progress in object detection [95] and feature matching [191], provides robust and accurate pose estimations on our in-the-wild videos, which is the key to the success of our framework. Our model trained on the Accidental Turntables dataset achieves higher pose prediction accuracy than when trained on the FreiburgCars dataset.

In comparison with analysis-by-synthesis frameworks [143, 155], our prediction accuracy is significantly higher than that of SSV model [155] which is trained on the CompCars dataset [272] (consisting of 137,000 real car images). ViewNet [143] achieves the highest performance on PASCAL3D+ among existing unsupervised learning methods. However, this method relies on 3D models from ShapeNet [28] to generate a highly curated dataset with controlled variations in viewpoint, translation, lighting, background, etc. In contrast, ViewNet has a harder time learning from real videos (*e.g.*, FreiburgCars [197]) where its performance drops remarkably.

Qualitative results. Fig. 3.9 visualizes our pose predictions on the Pascal3D+ test set. Our model provides accurate pose estimation on diverse cars in terms of appearance, poses, and shapes. The performance of our model drops in several cases: the object is highly occluded; the image is in low resolution; the domain gap between the input and our dataset is large (*e.g.*, cartoon cars, snow-covered cars). These issues can be potentially relieved by collecting more videos to further enrich the diversity of cars in our dataset.

Figure 3.9: Pose prediction on Pascal3D+ test set. Left: our model achieves high accuracy of pose estimation on cars in diverse appearances, poses, and shapes. Right: the performance drops on large, occluded objects (1st row), low-resolution images (2nd row) or out-of-domain data (last two rows). The solid arrows indicate the pose predictions from our model and the dashed arrows are the groundtruth annotations. The blue arrow directs towards the frontal side of cars and the red points toward the right side. The angular distances between the predictions and the groundtruth are less than 7° for examples on the left while higher than 90° on the failure cases.

3.2.5.3 Analysis

The emergence of a canonical pose. The key to the success of the proposed model is the emergence of the canonical pose in our first training stage. Fig. 3.10 provides images from our dataset with similar pose annotations after the calibration step (Sec. 3.2.4). On the one hand, Fig. 3.10 clearly demonstrates that the calibrated pose annotations align well in a uniform frame. On the other hand, the calibration

Figure 3.10: Canonical pose emerges in our first training stage (Sec. 3.2.4). For each reference image (top), we present four matches (including one failure case) of which the pose annotations have less than 5° angular distance to that of the reference frame. The calibration error $\mathcal{L}^*_{\text{cali}}$ (Eqn. 3.9) is higher than 25° on these failure cases while lower than 10° on the well-calibrated video instances. This provides us with a heuristic to filter out noisy annotations.

fails on several videos due to the limited performance of our stage-one model (Fig. 3.10 bottom). A typical failure case is that the pose predictor misidentifies the frontal view of a car as the rear view. Such failure cases of pose calibration introduce noisy pose annotations into our dataset. Fortunately, we find that the noise level of the annotations is closely correlated with the calibration error $\mathcal{L}^*_{\text{cali}}$ (Eqn. 3.9). We thus use the calibration error $\mathcal{L}^*_{\text{cali}}$ as a heuristic to filter out noisy annotations in our second training stage. We provide a detailed analysis below.

The effect of the noise level in the annotations. We use the calibration error $\mathcal{L}^*_{\text{cali}}$ (Eqn. 3.9) as an indicator of the noise level of the pose annotations. A higher threshold on the calibration error corresponds to a larger number of training images yet more noisy annotations, and vice versa. Fig. 3.11 presents the performance of our model under different noise levels of the annotations. It demonstrates that neither

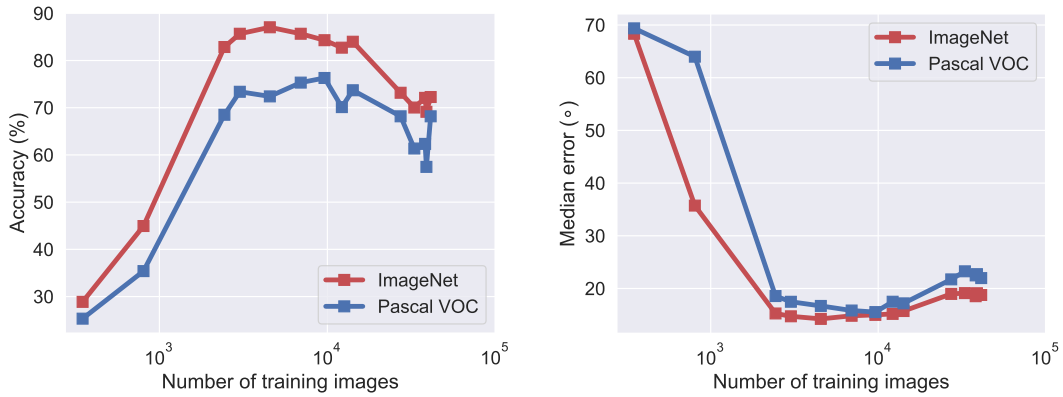


Figure 3.11: The effect of annotation noise level on 3D pose prediction. We report the performance of our pose estimation model under different noise levels of pose annotations. A higher level of annotation noise corresponds to a larger number of training images. We report both prediction accuracy (left panel) and median error (right panel) on two test splits included in PASCAL3D+.

clean-yet-small data nor large-yet-noisy data lead to higher performance than mid-size data with mid-level noise.

The effect of two-stage training. As demonstrated in Fig 3.10, the model trained in the first stage provides a tool to calibrate the pose annotations of our dataset. However, the performance of the stage-one model lags behind the state-of-the-art analysis-by-synthesis frameworks (*e.g.*, SSV [155] and ViewNet [143]). We hypothesize that training to predict the relative pose is a suboptimal learning strategy for the task of absolute pose estimation. As shown in Tab. 3.7, the model trained in our second training stage significantly outperforms the one trained in the first stage. This suggests that learning with absolute pose annotations is a more effective training method. However, our stage-two training is not possible without the pose calibration and stage-one model. Therefore, the proposed two training stages are complementary and both play an important role in our framework.

The effect of network initialization. The recent self-supervised learning (SSL) [36, 94] has significantly improves the unsupervised pose estimation [38] and part discovery [186]. We initialize our pose estimation network with ImageNet-pretrained

Table 3.7: The effect of two-stage training on 3D pose prediction. The second stage trains the model to regress to absolute pose after using the first stage model to calibrate the relative pose annotations. This procedure leads to a significant improvement in pose estimation accuracy (%) and median error (°), in spite of the training datasets.

Trainset	Stage	PASCAL VOC		ImageNet	
		Acc. ↑	Med. ↓	Acc. ↑	Med. ↓
Acci. Turn.	1	42	38.8	46	32.9
	2	**75**	**15.8**	**86**	**14.8**
FreiburgCars	1	36	44	47	31.9
	2	**72**	**15.7**	**84**	**15.0**

Table 3.8: The effect of network initialization on 3D pose prediction. ImageNet pretrained models provide a significant improvement over random initialized ones but self-supervised counterparts are competitive alternatives without having to resort to extra human annotations.

Initialization	PASCAL VOC		ImageNet	
	Acc. ↑	Med. ↓	Acc. ↑	Med. ↓
Random	58	25	70	20.2
Contrastive [94]	74	15.7	85	**14.3**
ImageNet	**75**	**15.8**	**86**	14.8

models by default. However, ImageNet classification labels require extensive human labor. A natural question is how the recent SSL methods help us further reduce the requirement of human annotations. Tab. 3.8 provides a comparison of different initialization strategies. Supervised ImageNet pretraining and unsupervised contrastive pretraining [36, 94] have similar performance in the task of pose estimation, while both outperform the random initialization in a large margin.

Pose distribution. Figure 3.12 compares the pose distribution of the Acci- dental Turntables dataset and PASCAL3D+. The distribution of azimuth is more balanced in our dataset, where PASCAL3D+ has more cars with large elevations.

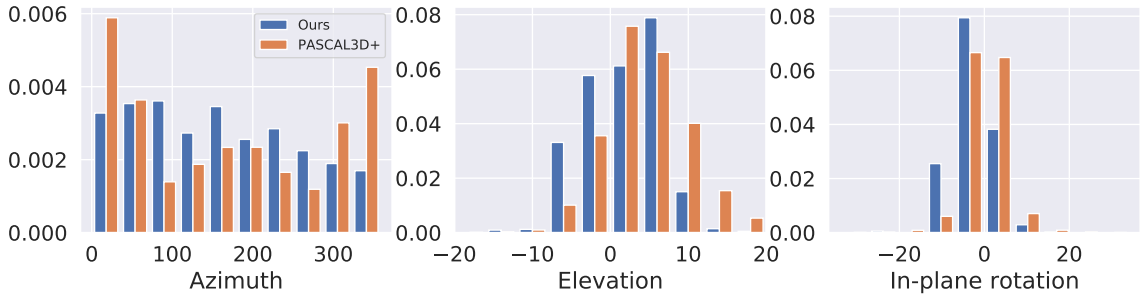


Figure 3.12: Distribution of the poses in the proposed Accidental Turntables dataset and the PASCAL3D+.

Feature extraction and matching for SfM. Feature extraction and matching are the core of SfM algorithms. The classical SIFT [138] and simple nearest neighbor matching (NN) remain the default components in popular SfM packages (*e.g.*, COLMAP [194, 195]), despite of the recent success of learning-based methods [56, 191]. We observe that SfM with SIFT and NN does not work reliably on our in-the-wild video dataset. Fig. 3.13 compares the 3D reconstruction and pose estimation from COLMAP under different feature extraction and matching algorithms on two videos from our dataset. SfM with SIFT and NN only provides partial 3D reconstruction and pose estimation on a small subset of frames. Its performance drops significantly on texture-free objects (Fig. 3.13 bottom). Simply replacing SIFT with Superpoint [56] leads to more complete 3D reconstruction and pose estimations. SfM with Superpoint and SuperGlue [191] provides the highest quality of shape reconstruction and pose estimations. Our experimental results can be explained by the following observations: SIFT detects few interest points on most cars due to the texture-free surface; SIFT extracts feature in a small local region, which results in large ambiguity in matching duplicated patterns (*e.g.*, frontal and rear wheels of a car); large motion blur further destabilizes the feature-matching process; In comparison, Superpoint provides rich interest points even in texture-free regions; Lastly, SuperGLUE aggregates long-range contextual information via an attention mechanism, which we

find significantly reduces the ambiguity in matching repeated patterns. Fig. 3.14 provides more examples from our Accidental Turntables dataset. The performance of SfM may drop on highly-occluded objects (*e.g.*, the car is occluded by smoke in Fig. 3.14 bottom).

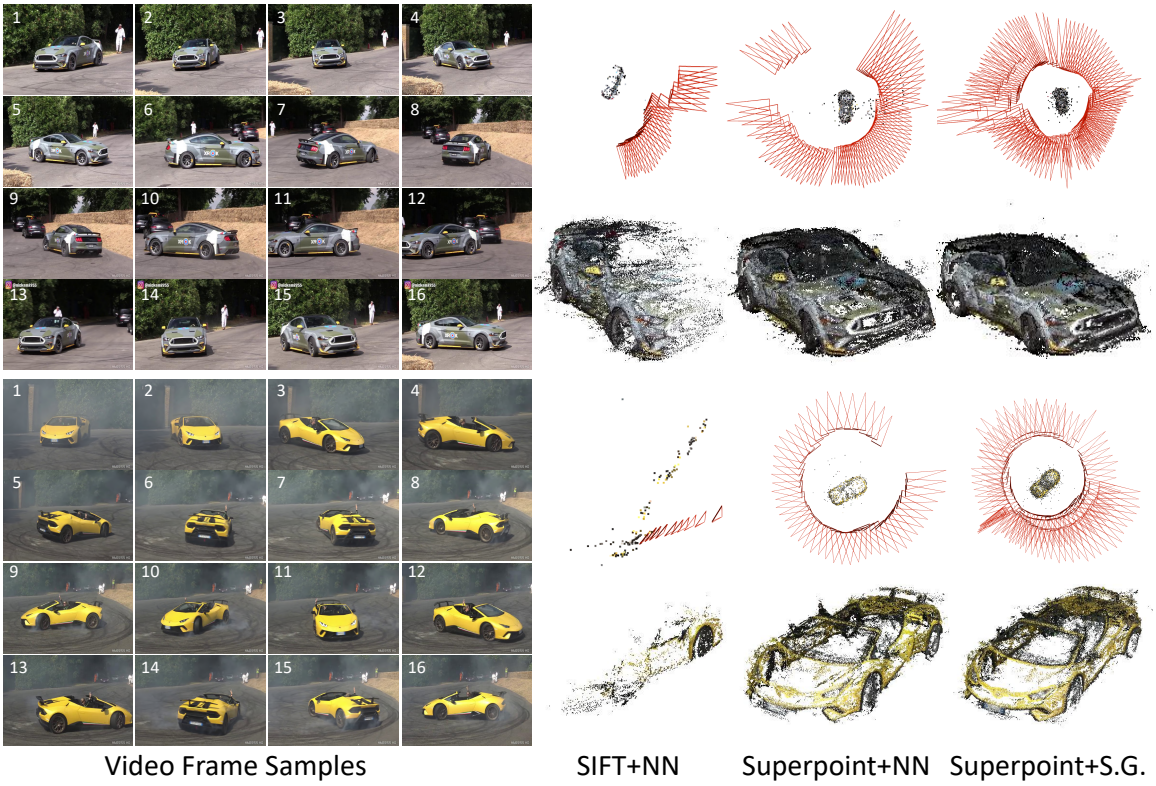


Figure 3.13: Feature extraction and matching for structure-from-motion. Left: video samples from the proposed Accidental Turntables dataset. Right: pose estimations (top) and dense 3D reconstruction (bottom) under different feature extraction (SIFT [138] or Superpoint [56]) and matching (nearest neighbor (NN) or SuperGlue (S.G.) [191]) algorithms. The red square pyramids indicate the location of the estimated camera pose. Each video consists of more than 200 frames and the car turns around 720°.

Extension to other categories. There are a fair number of turntable videos for other categories on Youtube. For example, airplanes turn along the runway (*e.g.*, video1, video2); landing or takeoff of airplanes usually induces more than 90-degree pose changes relative to the camera (*e.g.*, video3, video4); cruises turn (*e.g.*, video7). Fig. 3.15 shows SfM with Superpoint, and SuperGlue provides reasonable pose esti-

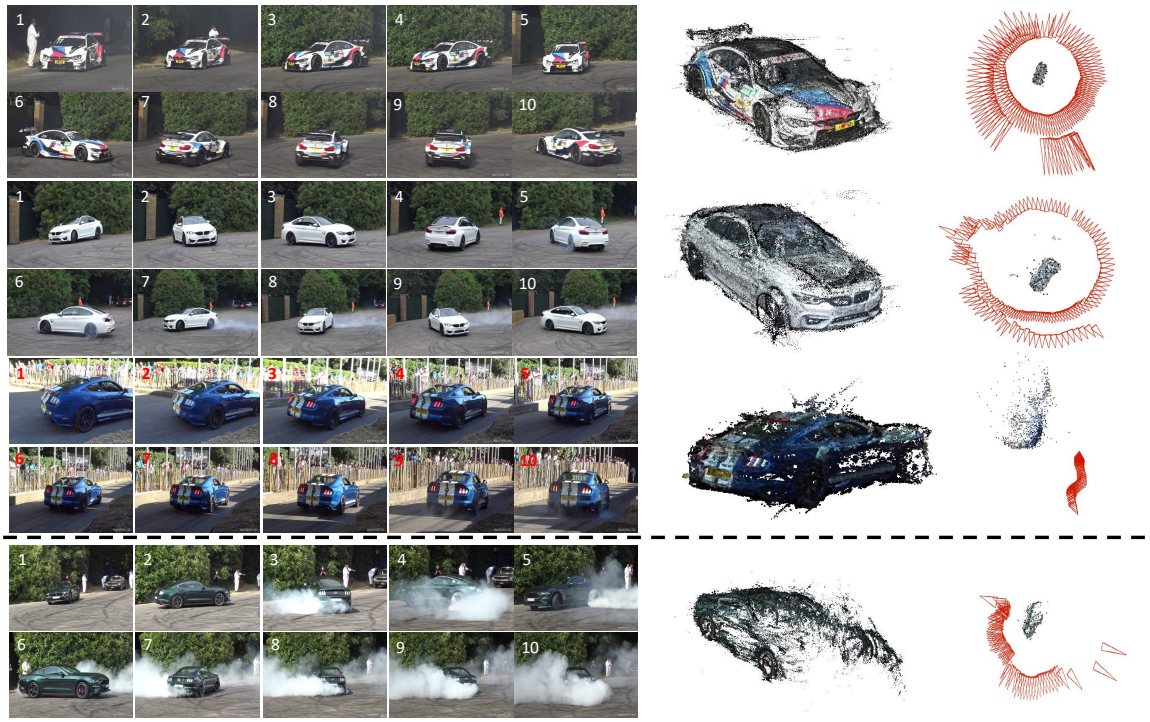


Figure 3.14: More examples from the Accidental Turntables dataset. SfM provides accurate 3D reconstructions and pose estimations on either texture-rich (1st row) or texture-free (2nd row) objects, as well as objects moving along a straight line without any turns (3rd row). The performance drops on highly-occluded objects (bottom).

Figure 3.15: Accidental Turntables for airplanes and cruise. Left: video frame samples. Right: pose estimation and 3D reconstruction from structure-from-motion.

mation and 3D reconstruction on these categories. Even though we focus on cars in this work, our dataset is much larger, easier to collect, and more useful to train a pose estimator than existing car datasets (*e.g.*, FreiburgCars).

3.2.6 Conclusion and subsequent works

We propose to learn 3D pose estimation models from a new source of data: videos where objects turn. We demonstrate that classical structure-from-motion algorithms, coupled with the recent advances in feature matching and object detection, provide surprisingly accurate pose estimations and 3D reconstructions on in-the-wild car videos. We also provide a novel learning framework that successfully trains a high-quality 3D pose predictor on the collected video datasets.

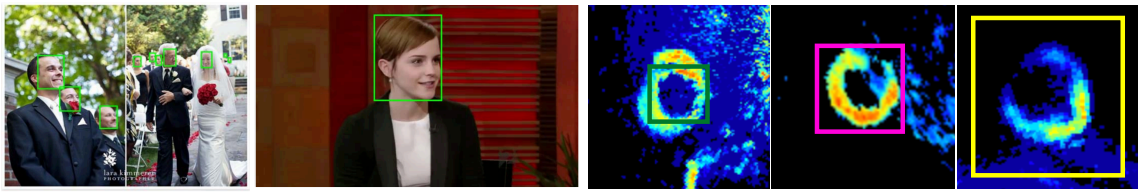
Subsequent to this work, more recently, we propose a method to jointly estimate 3D scene representation and camera poses from a collection of unposed images or in-the-wild videos [223] (see Chapter 6 for more details). This method demonstrates complementary performance to the SfM pipeline adopted in our current work. Future research will aim to generalize the proposed method to encompass a wider range of categories, as well as explore the feasibility of training category-agnostic models for 3D pose estimation.

CHAPTER 4

LEARNING FROM HETEROGENEOUS LABELS

Besides the prohibitive annotation cost, another common issue regarding human annotations is the label noise, which may significantly hurt the performance of machine learning models [157]. A common practice of collecting labels is to distribute the annotation task to many annotators through online tools such as Amazon Mechanical Turk. However, different annotators may have quite different understandings of the annotation policy. This could lead to inconsistent annotations, resulting in what are referred to as *heterogeneous labels*.

One way to relieve the issue of heterogeneous labels is to collect multiple annotations per instance from different annotators and filter labels based on the agreement among the annotators [127]. However, this increases annotation costs and is not applicable when the annotations are collected opportunistically across different studies without consistent annotation guidelines. For example, Jiang *et al.* [107] observed that the style of the bounding box annotations for human faces changes dramatically across different benchmarks — the face annotations from the IJB-A benchmark [114] include the whole head while the annotations from WIDER [273] exclude the top of the head (see Fig. 4.1a). A similar issue appears in the bird roost annotations on the weather radar images, collected from different researchers and naturalists in prior research studies [118] (see Fig. 4.1b and Sec. 4.2). This annotation variation makes evaluation using held-out data very difficult and inhibits learning due to inconsistent supervision.



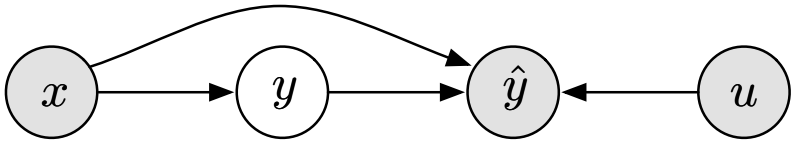
(a) WIDER and IJB-A face dataset (b) Tree Swallow roost dataset

Figure 4.1: Heterogeneous labels. (a) shows the face annotations from WIDER (left) [273] and IJB-A benchmark (right) [114]. Examples are taken from the work of Jiang *et al.* [107]. (b) presents the Tree Swallow roost annotations (*i.e.*, the ring-like patterns) from three different annotators. Observe the variations in the tightness of bounding boxes across different benchmarks and annotators.

In this chapter, we consider such a specific type of noisy annotation that is caused by the variations of labeling styles across annotators or benchmarks, dubbed heterogeneous labels. Despite the sheer volume of prior works on learning from noisy labels (see Sec. 2.2), learning from heterogeneous labels is yet under-explored in the literature. Sec. 4.1 presents a principled learning method, and Sec. 4.2 introduces a novel application of our framework in an ecological study. We focus on the object detection task in this chapter and leave its extension to other tasks as our future work.

4.1 Approach

Our goal is a generic and principled approach that can leverage standard detection frameworks (*e.g.*, Faster RCNN [178]) with little or no modification. To model variability due to annotation styles we use the following graphical model:



where x is the image, y represents the unobserved "true" or gold-standard label, u is the user (or features thereof), and $\hat{y}$ is the observed label in user u's labeling style. In this model

- $p_\theta(y|x)$ is the *detection model*, with parameters θ. We generally assume the negative log-likelihood of the detection model is equal to the loss function of the base detector. For example, in our application, $-\log p_\theta(y|x) = L_{\text{cnn}}(\theta|y)$, the loss function of Faster R-CNN.[1]
- $p_\beta(\hat{y} \mid x, y, u)$ is the *forward user model* for the labeling style of user u, with parameters β. In our application, much of the variability can be captured by user-specific scaling of the bounding boxes, so we adopt the following user model: for each bounding box, we model the observed radius as $p_\beta(\hat{r} \mid r, u) = \mathcal{N}(\hat{r}; \beta_u r, \sigma^2)$ where r is the unobserved true radius and β_u is the user-specific scaling factor. In this model, the bounding-box centers are unmodified and the user model does not depend on the image x, even though our more general framework allows both.
- $p_{\theta,\beta}(y \mid x, \hat{y}, u)$ is the *reverse user model*. It is determined by the previous two models and is needed to reason about the true labels given the noisy ones during training. Since this distribution is generally intractable, we use instead a variational reverse user model $q_\phi(y \mid x, \hat{y}, u)$, with parameters ϕ. In our application, $q_\phi(r \mid \hat{r}, u) = \mathcal{N}(r; \phi_u \hat{r}, \sigma^2)$, which is another user-specific rescaling of the radius.

We train the user models jointly with Faster R-CNN using variational EM. We initialize the Faster R-CNN parameters θ by training for 50K iterations starting from the ImageNet pretrained VGG-M model using the original uncorrected labels. We then initialize the forward user model parameters β using the Faster R-CNN predictions: if a predicted roost with radius r_i overlaps sufficiently with a labeled roost (intersection-over-union > 0.2) and has a high enough detection score (> 0.9), we

[1]Faster R-CNN includes a region proposal network to detect and localize candidate objects and a classification network to assign class labels. The networks share parameters and are trained jointly to minimize a sum of several loss functions; we take the set of all parameters as θ and the sum of loss functions as $L_{\text{cnn}}(\theta|y)$.

generate a training pair $(r_i, \hat{r}_i)$ where $\hat{r}_i$ is the labeled radius. We then estimate the forward regression model parameters as a standard linear regression with these pairs.

After initialization, we repeat the following steps (in which i is an index for annotations):

- Update parameters ϕ of the reverse user model by minimizing the combined loss $\mathbb{E}_{r_i \sim q_\phi(r_i|\hat{r}_i,u_i)}\left[L_{\text{cnn}}(\theta|\{r_i\}) - \sum_i \log p_\beta(\hat{r}_i|r_i,u_i)\right]$. The optimization is performed separately to determine the reverse scaling factor ϕ_u for each user using Brent's method with search boundary $[0.1, 2]$ and black-box access to L_{cnn}.
- Resample annotations on the training set by sampling $r_i \sim q_\phi(\cdot|\hat{r}_i, u_i)$ for all i, then update θ by training Faster R-CNN for 50K iterations using the resampled annotations.
- Update β by training the forward user models using pairs $(r_i, \hat{r}_i)$, where r_i is the radius of the imputed label.

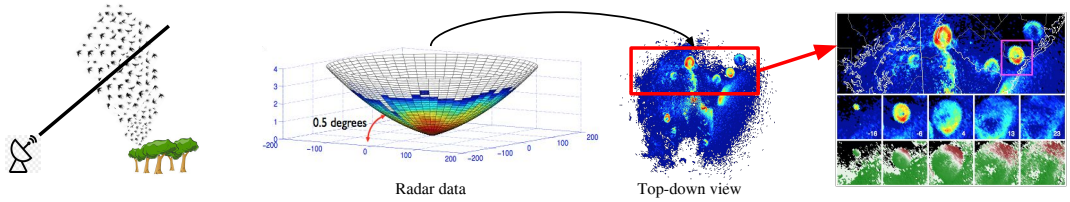
Formally, each step can be justified as maximizing the *evidence lower bound* (ELBO) [20] of the log marginal likelihood $\log p_{\theta,\beta}(\hat{y}|x,u) = \log \int p_{\theta,\beta}(\hat{y}, y|x, u)dy$ with respect to the variational distribution q_ϕ. Steps 1, 2, and 3 maximize the ELBO with respect to ϕ, θ, and β, respectively. Steps 1 and 2 require samples from the reverse user model; we found that using the *maximum a posteriori* y instead of sampling is simple and performs well in practice, so we used this in our application.

We assume y is a *structured* label that includes all bounding boxes for an image. This justifies equating $-\log p_\theta(y|x)$ with the loss function $L(\theta)$ of an existing detection framework that predicts bounding boxes simultaneously for an entire image (e.g., using heuristics like non-maximum suppression). This is important because it is modular. We can use any detection framework that provides a loss function with no other changes. A typical user model will then act on y (a set of bounding boxes) by acting independently on each of its components, as in our application.

We anticipate this framework can be applied to a range of applications. More sophisticated user models may also depend on the image x to capture different labeling biases, such as different thresholds for labeling objects or tendencies to mislabel objects of a certain class or appearance. However, it is an open question of how to design more complex user models and we caution about the possibility of very complex user models "explaining away" true patterns in the data.

4.2 Application: detecting and tracking Tree Swallow roosts

We evaluate our framework in a novel application of computer vision in ecology — detecting and tracking Tree Swallow roosts in weather radar data. We provide a background of this application in Sec. 4.2.1, introduce a detection and tracking system in Sec. 4.2.2, and demonstrate that our approach significantly improves the performance of this system trained on heterogeneous annotations in Sec. 4.2.3, which enables a series of large-scale ecological studies, as described in Sec. 4.2.4.



(a) Communal roost exodus (b) Radar geometry and roost appearance

Figure 4.2: Radar background. (a) Illustration of roost exodus. (b) A radar traces out cone-shaped slices of the atmosphere (left), which are rendered as top-down images (center). This image from the Dover, DE radar station at 6:52 am on Oct 2, 2010 shows at least 8 roosts. Several are shown in more detail to the right, together with crops of one roost from five consecutive reflectivity and radial velocity images over a period of 39 minutes. These show the distinctive expanding ring and "red-white-green" diverging velocity patterns.

4.2.1 Background

Radar Data. We use radar data from the US NEXRAD network of over 140 radars operated by the National Weather Service [48]. They have ranges of several hundred

kilometers and cover nearly the entire US. Data is available from the 1990s to the present in the form of raster data products summarizing the results of radar *volume scans*, during which a radar scans the surrounding airspace by rotating the antenna 360° at different elevation angles (e.g., 0.5°, 1.5°) to sample a cone-shaped "slice" of airspace (Fig. 4.2b). Radar scans are available every 4–10 minutes at each station. Conventional radar images are top-down views of these sweeps; we will also render data this way for processing.

Standard radar scans collect 3 data products at 5 elevation angles for 15 total channels. We focus on data products that are most relevant for detecting roosts. *Reflectivity* is the base measurement of the density of objects in the atmosphere. *Radial velocity* uses the Doppler shift of the returned signal to measure the speed at which objects are approaching or departing the radar. *Copolar cross-correlation coefficient* is a newer data product, available since 2013, that is useful for discriminating rain from biology [209]. We use it for post-processing, but not training, since most of our labels are from before 2013.

Roosts. A roost exodus (Fig. 4.2a) is the mass departure of a large flock of birds from a nighttime roosting location. They occur 15–30 minutes before sunrise and are very rarely witnessed by humans. However, roost signatures are visible on the radar as birds fly upward and outward into the radar domain. Fig. 4.2b, center, shows a radar reflectivity image with at least 8 roost signatures in a 300×300km area. Swallow roosts, in particular, have a characteristic signature shown in Fig. 4.2b, right. The center row shows reflectivity images of one roost expanding over time. The bottom row shows the characteristic radial velocity pattern of birds dispersing away from the center of the roost. Birds moving toward the radar station (bottom left) have negative radial velocity (green) and birds moving away from the radar station (top right) have positive radial velocity (red).

Annotations. We obtained a data set of manually annotated roosts collected for prior ecological research [118]. They are believed to be nearly 100% Tree Swallow roosts. Each label records the position and the radius of a circle within a radar image that best approximates the roost. We restricted to seven stations in the eastern US and to month-long periods that were exhaustively labeled, so we could infer the absence of roosts in scans with no labels. We restricted scans from 30 minutes before to 90 minutes after sunrise, leading to a data set of 63691 labeled roosts in 88972 radar scans. A significant issue with this data set is systematic differences in labeling style by different researchers. This poses serious challenges to building and evaluating a detection model.

Roost detection and tracking. There is a long history to the study of roosting behavior with the radar data, almost entirely based on human interpretation of images [23, 118, 252]. That work is therefore restricted to analyzing only limited regions, short-time periods, or coarse-grained information about the roosts. Chilson *et al.* [39] developed a deep-learning image classifier to identify radar images that contain roosts. While useful, this provides only limited biological information.

4.2.2 A roost detection and tracking system

Our overall approach consists of four steps (see Fig. 4.3): we render radar scans as multi-channel images, run a single-frame detector, assemble and rescore tracks, and then post-process detections using other geospatial data to filter specific sources of false positives.

Detection architecture. Our single-frame detector is based on Faster R-CNNs [178]. Region-based CNN detectors such as Faster R-CNNs are state-of-the-art on several object detection benchmarks.

A significant advantage of these architectures comes from pretraining parts of the network on large labeled image datasets such as ImageNet [53]. To make radar data

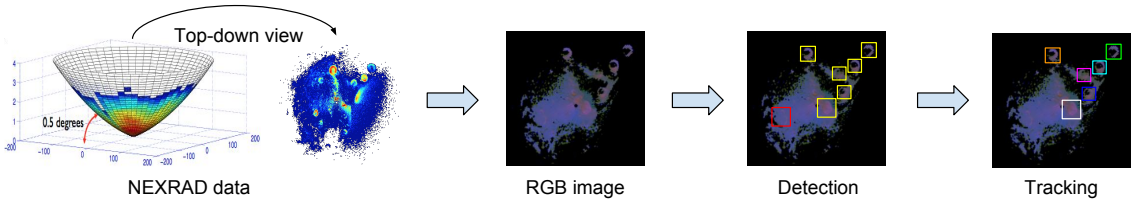


Figure 4.3: Detection and tracking pipeline. A final step (not shown) uses auxiliary data to filter rain and wind farms.

compatible with these networks, we must select only 3 of the 15 available channels to feed into the RGB-based models. We select the radar products that are most discriminative for humans: reflectivity at 0.5°, radial velocity at 0.5° degrees, and reflectivity at 1.5°. Roosts appear predominantly in the lowest elevations and are distinguished by the ring pattern in reflectivity images and distinctive velocity pattern. These three data products are then rendered as a 1200×1200 image in the "top-down" Cartesian-coordinate view (out to 150km from the radar station) resulting in a 3-channel 1200×1200 image. The three-channel images are fed into Faster R-CNN initialized with a pretrained VGG-M network [29]. All detectors are trained for the single "roost" object class, using bounding boxes derived from the labeled dataset described above.

Although radar data is visually different from natural images, we found ImageNet pretraining is quite useful; without pretraining the networks took significantly longer to converge and resulted in a 15% lower performance. We also experimented with models that map 15 radar channels down to 3 using a learned transformation. These networks were *not consistently better* than ones using hand-selected channels. Models trained with shallower networks that mimic handcrafted features, such as those based on gradient histograms, performed 15-20% worse depending on the architecture.

Training details. Preliminary experiments revealed that systematic variations in labeling style were a significant barrier to training and evaluating a detector. Fig. 4.4 shows example detections that correctly locate and circumscribe the ring-like pat-

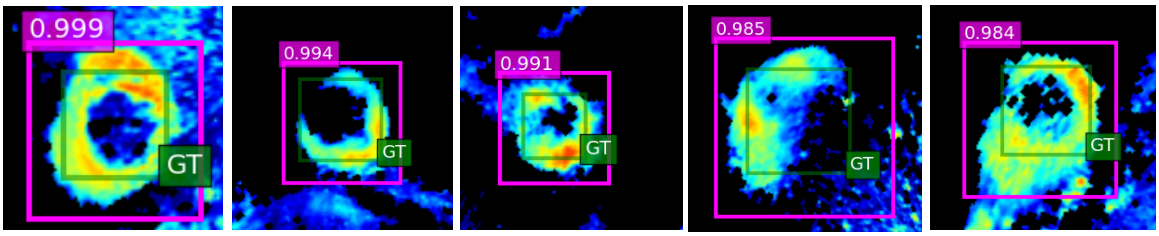


Figure 4.4: Labeling style variation leads to inaccurate evaluation and suboptimal detectors. All of these detections (pink boxes) are misidentified as false positives because of insufficient overlap with annotations of one user (green boxes) with a tight labeling style. Label variation also hurts training and leads to suboptimal models.

terns in the weather radar images (*i.e.*, Tree Swallow roosts, see for Sec. 4.2 more details). but are classified as false positives because the annotator used labels (originally circles) to "trace" roosts instead of circumscribing them. Although it is clear upon inspection that these detections are "correct", with 63691 labels and a range of labeling styles, there is no simple adjustment to accurately judge the performance of a system. Furthermore, labeling variation also inhibits learning and leads to suboptimal models. This motivates our approach to jointly learn a detector along with user-specific models of labeling style, as described in Sec. 4.1. We provide a detailed ablation study that shows the effectiveness of our training method in Sec. 4.2.3.

Roost tracking and rescoring. Associating and tracking detections across frames is important for several reasons. It helps rule out false detections due to rain and other phenomena that have different temporal properties than roosts. Detection tracks are also associated directly with the biological entity—a single flock of birds—so they are needed to estimate biological parameters such as roost size, rate of expansion, location, and habitat of first appearance, etc. We employ a greedy heuristic to assemble detections from individual frames into tracks [179], starting with high-scoring detections and incrementally adding unmatched detections with high overlap in nearby frames. Detections that match multiple tracks are assigned to the longest one. After associating detections, we apply a Kalman smoother to each track using a linear dy-

namical system model for the bounding box center and radius. This model captures the dynamics of roost formation and growth with parameters estimated from ground-truth annotations. We then conduct a final rescoring step where track-level features (*e.g.*, number of frames, average detection score of all bounding boxes in track) are associated with individual detections, which are then rescored using a linear SVM. This step suppresses false positives that appear roost-like in single frames but do not behave like roosts.

Postprocessing with auxiliary information. In preliminary experiments, the majority of high-scoring tracks were roosts, but there were also a significant number of high-scoring false positives caused by specific phenomena, especially wind farms and precipitation. We found it was possible to reliably reject these false positives using auxiliary information. To eliminate rain in modern data, we use the radar measurement of the copolar cross-correlation coefficient, ρ_{HV}, which is available since 2013 [209]. Biological targets have much lower ρ_{HV} values than precipitation due to their high variance in orientation, position and shape over time. A common rule is to classify pixels as rain if $\rho_{HV} > 0.95$ [58]. We classify a roost detection as precipitation if a majority of pixels inside its bounding box have $\rho_{HV} > 0.95$. For historical data one may use automatic methods for segmenting precipitation in radar images such as [129]. For wind farms, we can use recorded turbine locations from the U.S. Wind Turbine Database [103]. A detection is identified as a wind farm if any turbine from the database is located inside its bounding box.

4.2.3 Experiments

Dataset. We divided the 88972 radar scans from the manually labeled dataset into training, validation, and test sets. Tab. 4.1 gives details of training and test data by the station. The validation set (not shown) is roughly half the size of the test set and was used to set the hyper-parameters of the detector and the tracker.

Evaluation metric. To evaluate the detector we use established evaluation metrics for object detection employed in common computer vision benchmarks. A detection is a true positive if its overlap with an annotated bounding box, measured using the intersection-over-union (IoU) metric, is greater than 0.5. The mean average precision (MAP) is computed as the area under the precision-recall curve. For the purposes of evaluating the detector, we mark roosts smaller than 30×30 in a 1200×1200 radar image as difficult and ignore them during evaluation. Humans typically detect such roosts by looking at adjacent frames. As discussed previously (Fig. 4.4), evaluation is unreliable when user labels have different labeling styles. To address this, we propose an evaluation metric ("+User") that rescales predictions on a per-user basis prior to computing MAP. Scaling factors are estimated following the same procedure used to initialize variational EM. This assumes that the user information is known for the test set, where it is *only* used for rescaling predictions and not by the detector.

Results: roost detector and user model. Tab. 4.1 shows the performance of various detectors across radar stations. We trained two detector variants, one a standard Faster R-CNN, and another trained with the variational EM algorithm. We evaluated the detectors based on whether annotation bias was accounted for during testing (Tab. 4.1, "+User").

The noisy annotations cause inaccurate evaluation. A large number of the detections on `KDOX` are misidentified as negatives because of the low overlap with the annotations, which are illustrated in Fig. 4.4, leading to a low MAP score of 9.1%. This improves to 44.8% when the annotation biases are accounted for during testing. As a sanity check, we trained and evaluated a detector on annotations of a single user on `KDOX` and found its performance to be in the mid-fifties. However, the score was low when this model was evaluated on annotations from other users or stations.

The detector trained jointly with user-models using variational EM further improves performance across all stations (Tab. 4.1, "+EM+User"), with larger improve-

Station	Test	Train	R-CNN	+User	+EM+User
KMLB	9133	19998	47.5	47.8	**49.2**
KTBW	7195	16382	47.3	50.0	**50.8**
KLIX	4077	10192	32.4	35.1	**35.7**
KOKX	1404	2994	23.2	27.3	**29.9**
KAMX	860	1898	29.9	30.8	**31.6**
KDOX	639	902	9.1	44.8	**50.2**
KLCH	112	441	32.1	39.8	**43.1**
entire	23.7k	53.6k	41.0	44.2	**45.5**

Table 4.1: Roost detection MAP for detector variants. We use Faster RCNN [178] (dubbed "R-CNN") as our object detection model.

ments for stations with less training data. Overall MAP improves from 44.2% to 45.5%. To verify the statistical significance of this result, we drew 20 sets of bootstrap resamples from the entire test set (containing $23.7k$ images) and computed the MAP of the model trained with EM and without EM on each set. The mean and standard error of MAPs for the model trained with EM is 45.5% and 0.12% respectively, while they are 44.4% and 0.11% for the model trained without EM.

Results: tracking and rescoring. After obtaining the roost detections from our single-frame detector, we can apply our roost tracking model to establish roost tracks over time. Fig. 4.5 shows an example radar sequence where roost detections have been successfully tracked over time and some false positives removed. We also systematically evaluated the tracking and rescoring model on scans from the KOKX station. For this study, we performed a manual evaluation of the top 800 detections before and after the contextual rescoring. The manual evaluation was necessary due to human labeling biases, especially the omission of labels at the beginning or end of a roost sequence when roost signatures are not as obvious. Fig. 4.5, the middle panel, shows that the tracking and rescoring improve the precision across the entire range of k. Our tracking model also enables us to study the roost dynamics over time (see Sec. 4.2.4 and Fig. 4.5 right panel).

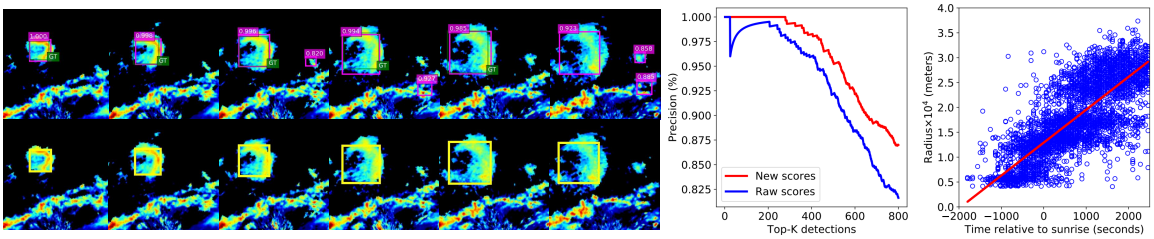


Figure 4.5: Roost tracking. Left: tracking example, with raw detections (top) and track (bottom). Transient false positives in several frames lead to poor tracks and are removed by the rescoring step. Middle: precision@k before and after rescoring. Right: Roost radius relative to time after sunrise.

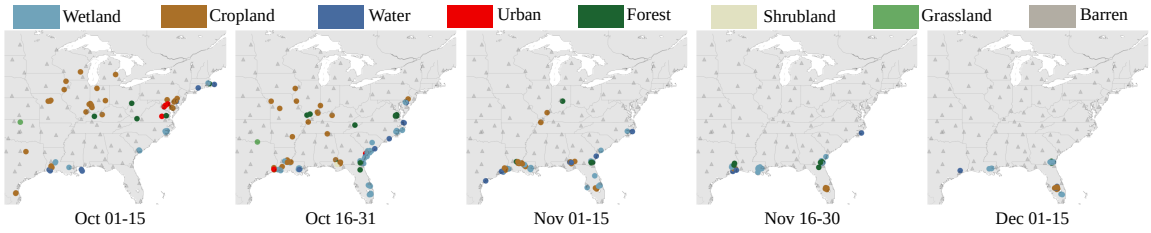


Figure 4.6: Tree Swallow fall migration in 2013. The color circles show detected roost locations with each half-month period. The location of each roost is determined by the center of the first bounding box in the track when the airborne birds are closest to their location on the ground. Faint gray triangles show radar station locations.

4.2.4 Case study

We conducted a case study to use our detector and tracker to synthesize knowledge about continent-scale movement patterns of swallows. We applied our pipeline to 419k radar scans collected from 86 radar stations in the Eastern US (see Figure 4.6) from October 2013 through March 2014. During these months, Tree Swallows are the only (fall/winter) or predominant (early spring) swallow species in the US and responsible for the vast majority of radar roost signatures. This case study is therefore the first system to obtain comprehensive measurements of a single species of bird across its range on a daily basis. We ran our detector and tracking pipeline on all radar scans from 30 minutes before sunrise to 90 minutes after sunrise. We kept tracks having at least two detections with a detector score of 0.7 or more and then ranked tracks by the sum of the detector score for each detection in the track.

	Pre	Post		Pre	Post
Swallow roost	454	449	Other roost	38	38
Precipitation	109	5	Clutter	22	21
Wind farm	47	0	Unknown	8	8

Table 4.2: Detections by type pre- and post-filtering with auxiliary data. Post-processing effectively removes false positives due to precipitation and wind farms.

Error analysis. There were several specific phenomena that were frequently detected as false positives prior to post-processing. We reviewed and classified all tracks with a total detection score of 5 or more prior to postprocessing (678 tracks total) to evaluate detector performance "in the wild" and the effectiveness of post-processing. This also served to vet the final data used in the biological analysis. Tab. 4.2 shows the number of detections by category before and after post-processing. Roughly two-thirds of initial high-scoring detections were swallow roosts, with another 5.6% being communal roosts of *some* bird species.

The most false positives were due to precipitation, which appears as highly complex and variable patterns in radar images, so it is common to find small image patches that share the general shape and velocity pattern of roosts (Fig. 4.7, fourth column). Humans recognize precipitation from larger-scale patterns and movement. Filtering using ρ_{HV} nearly eliminates rain false positives. The second leading source of false positives was wind farms. Surprisingly, these share several features of roosts: they appear as small high-reflectivity "blobs" and have a diverse velocity field due to spinning turbine blades (Fig. 4.7 last column). Humans can easily distinguish wind farms from roosts using temporal properties. *All* wind farms are filtered successfully using the wind turbine database. Since our case study focuses on Tree Swallows, we marked as "other roost" detections that were believed to be from other communally roosting species (*e.g.*, American Robins, blackbirds, crows). These appear in radar less frequently and with a different appearance (usually "blobs" instead of "rings";

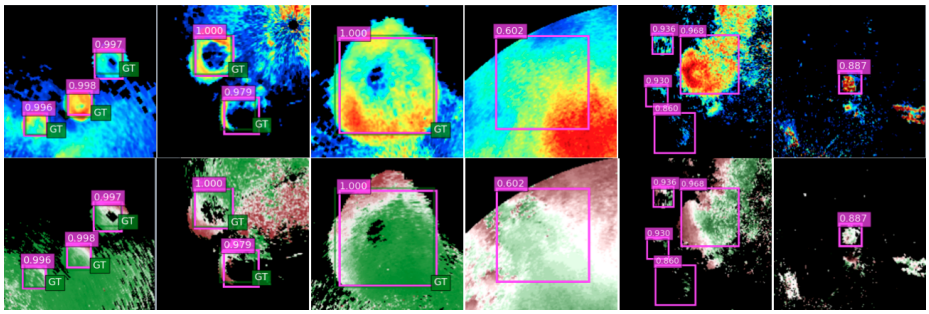


Figure 4.7: Visualization of roost detections. Some detections are visualized on the reflectivity (top) and radial velocity (bottom) channels of different scans. The first three columns show swallow roost detections while the next three columns show detections due to rain, roosts of other species, and windmills.

Fig. 4.7, fifth column) due to behavioral differences. Humans use appearance cues as well as habitat, region, and time of year to judge the likely species of a roost. We marked uncertain cases as "other roost".

Migration and habitat use. Fig. 4.6 shows swallow roost locations and habitat types for five half-month periods starting in October to illustrate the migration patterns and seasonal habitat use of Tree Swallows. Habitat assignments are based on the majority of habitat classes from the National Land Cover Database (NLCD) [238] within a 10×10km area surrounding the roost center, following the approach of [23] for Purple Martins. Unlike Purple Martins, the dominant habitat type for Tree Swallows is wetlands (38% of all roosts), followed by croplands (29%). These reflect the known habits of Tree Swallows to roost in reedy vegetation—either natural wetlands (*e.g.*, cattails and phragmites) or agricultural fields (e.g., corn, sugar cane) [253].

In early October, Tree Swallows have left their breeding territories and formed migratory and pre-migratory roosts throughout their the breeding range across the northern US [253]. Agricultural roosts are widespread in the upper Midwest. Some birds have begun their southbound migration, which is evident by the presence of roosts along the Gulf Coast, which is outside the breeding range. In late October, roosts concentrate along the eastern seaboard (mostly wetland habitat) and in the

central US (mostly cropland). Most of the central US roosts occur near major rivers (e.g., the Mississippi) or other water bodies. The line of wetland roosts along the eastern seaboard likely delineates a migration route followed by a large number of individuals who make daily "hops" from roost to roost along this route [252]. By early November, only a few roosts linger near major water bodies in the central US. Some birds have left the US entirely to points farther south, while some remain in staging areas along the Gulf Coast [118]. By December, Gulf Coast activity has diminished, and roosts concentrate more in Florida, where a population of Tree Swallows will spend the entire winter.

Widespread statistics of roost locations and habitat usage throughout a migratory season has not previously been documented but are enabled by our AI system to automatically detect and track roosts. Our results are a starting point for better understanding and conserving these populations. They highlight the importance of the eastern seaboard and Mississippi valley as migration corridors, with different patterns of habitat use (wetland vs. agricultural) in each. The strong association with agricultural habitats during the harvest season suggests interesting potential interactions between humans and the migration strategy of swallows.

Roost emergence dynamics. Our AI system also enables us to collect more detailed information about roosts than previously possible, such as their dynamics over time to answer questions about their behavior. Fig. 4.5 shows the roost radius relative to time after sunrise for roosts detected by our system. Roosts appear around 1000 seconds before sunrise and expand at a fairly consistent rate. The best-fit line corresponds to swallows dispersing from the center of the roost with an average airspeed velocity of $6.61\,\mathrm{m\,s^{-1}}$ (unladen).

4.3 Conclusion and subsequent works

In this chapter, we show that user-specific label noise is a significant hurdle to doing machine learning with the available data set, and present a principled approach to overcome this. We demonstrate the effectiveness of our approach in a novel application of computer vision in ecology — detecting communal bird roosts using weather radar. Our system reveals new insights into the continental-scale roosting behavior of migratory Tree Swallows.

Building upon this work, we conduct a historical analysis using 20+ years of archived radar data to study the long-term bird population patterns in comparison with climate and land use change in the Great Lakes region of the US [17, 55]. We also incorporate spatial and temporal information using novel adaptor layers into the detection model, further improving the accuracy of our system and extending to other species such as bats [174]. As an ongoing project, we are collaborating with ecologists to collect more fine-grained measurements of birds — across the continent, at a daily time scale — from the entire 20-year radar archive. We will hopefully gain some of the first insights into the ecosystem and conservation.

CHAPTER 5

A BAYESIAN PERSPECTIVE ON NEURAL NETWORKS

In the deep learning era, one fundamental research problem is how to design neural network architectures that are efficient and effective for the target tasks (*e.g.*, image classification, object detection). Despite the sheer volume of existing works on this topic [59, 96, 178, 180, 204, 216], designing network architectures highly relies on extensive trial and error. To design the networks in a principled manner, it is necessary to understand the neural networks theoretically.

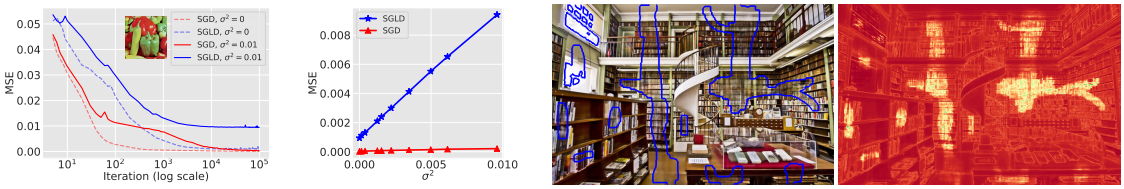
In this chapter, we provide a theoretical interpretation of neural network architectures through the lens of Gaussian processes, to demystify the surprising performance of *deep image prior* [237] in image restoration tasks. The deep image prior was introduced as a prior for natural images. It represents images as the output of a convolutional network with random inputs. For "inference", gradient descent is performed to adjust network parameters to make the output match observations. This approach yields good performance on a range of image reconstruction tasks. We show that the deep image prior is asymptotically equivalent to a stationary Gaussian process prior in the limit as the number of channels in each layer of the network goes to infinity, and derive the corresponding kernel. This informs a Bayesian approach to inference. We show that by conducting posterior inference using stochastic gradient Langevin dynamics we avoid the need for early stopping, which is a drawback of the current approach, and improve results for denoising and impainting tasks. We illustrate these intuitions on a number of 1D and 2D signal reconstruction tasks.

5.1 Overview

It is well known that deep convolutional networks trained on large datasets provide a rich hierarchical representation of images. Surprisingly, several works have shown that convolutional networks with *random* parameters can also encode non-trivial image properties. For example, second-order statistics of filter responses of random convolutional networks are effective for style transfer and synthesis tasks [239]. On small datasets, features extracted from random convolutional networks can work just as well as trained networks [192]. Along these lines, the *"deep image prior"* proposed by Ulyanov *et al.* [237] showed that the output of a suitably designed convolutional network on random inputs tends to be smooth and induces a natural image prior, so that the search over natural images can be replaced by gradient descent to find network parameters and inputs to minimize a reconstruction error of the network output. Remarkably, no prior training is needed and the method operates by initializing the parameters randomly.

Our work provides a novel Bayesian view of the deep image prior. We prove that a convolutional network with random parameters operating on a stationary input, *e.g.*, , white noise, approaches a two-dimensional Gaussian process (GP) with a *stationary kernel* in the limit as the number of channels in each layer goes to infinity (Theorem 5.1). While prior work [21, 144, 156, 160, 250] has investigated the GP behavior of infinitely wide networks and convolutional networks, our work is the first to analyze the spatial covariance structure induced by a convolutional network on stationary inputs. We analytically derive the kernel as a function of the network architecture and input distribution by characterizing the effects of convolutions, non-linearities, up-sampling, down-sampling, and skip connections on the spatial covariance. These insights could inform choices of network architecture for designing 1D or 2D priors.

We then use a Bayesian perspective to address the drawbacks of current estimation techniques for the deep image prior. Estimating parameters in a deep network from



(a) MSE vs. Iteration (b) Final MSE vs. σ^2 (c) Inferred mean (d) Inferred variance

Figure 5.1: Denoising and inpainting results with the deep image prior. **(a)** Mean Squared Error (MSE) of the inferred image with respect to the noisy input image as a function of iteration for two different noise levels. SGD converges to zero MSE resulting in overfitting while SGLD roughly converges to the noise level in the image. This is also illustrated in panel **(b)** where we plot the MSE of SGD and SGLD as a function of the noise level σ^2 after convergence. See Section 5.4.2 for implementation details. **(c)** An inpainting result where parts of the image inside the blue boundaries are masked out and inferred using SGLD with the deep image prior. **(d)** An estimate of the variance obtained from posterior samples visualized as a heat map. Notice that the missing regions near the top left have lower variance as the area is uniform.

a single image poses a huge risk of overfitting. In prior work the authors relied on early stopping to avoid this. Bayesian inference provides a principled way to avoid overfitting by adding suitable priors over the parameters and then using posterior distributions to quantify uncertainty. However, posterior inference with deep networks is challenging. One option is to compute the posterior of the limiting GP. For small networks with enough channels, we show this closely matches the deep image prior but is computationally expensive. Instead, we conduct posterior sampling based on stochastic gradient Langevin dynamics (SGLD) [247], which is both theoretically well-founded and computationally efficient since it is based on standard gradient descent. We show that posterior sampling using SGLD avoids the need for early stopping and performs better than vanilla gradient descent on image denoising and inpainting tasks (see Figure 5.1). It also allows us to systematically compute variances of estimates as a measure of uncertainty. We illustrate these ideas on a number of 1D and 2D reconstruction tasks.

5.2 Bayesian interpretation of deep image prior

5.2.1 Limiting Gaussian Process for convolutional networks

Previous work focused on the covariance of (scalar-valued) network outputs for two different inputs (*i.e.*, images). For the deep image prior, we are interested in the *spatial* covariance structure within each layer of a convolutional network. As a basic building block, we consider a multi-channel input image X transformed through a convolutional layer, an elementwise non-linearity, and then a second convolution to yield a new multi-channel "image" Z, and derive the limiting distribution of a representative channel z as the number of input channels and filters go to infinity. First, we derive the limiting distribution when X is *fixed*, which mimics derivations from previous work. We then let X be a stationary random process and show how the spatial covariance structure propagates to z, which is our main result. We then apply this argument inductively to analyze multi-layer networks and also analyze other network operations such as upsampling, downsampling, *etc.*

5.2.2 Limiting distribution for fixed input

For simplicity, consider an image $X \in \mathbb{R}^{c \times T}$ with c channels and only one spatial dimension. The derivations are essentially identical for two or more spatial dimensions. The first layer of the network has H filters denoted by $U = (u_1, u_2, \ldots, u_H)$ where $u_k \in \mathbb{R}^{c \times d}$ and the second layer has one filter $v \in \mathbb{R}^H$ (corresponding to a single channel of the output of this layer). The output of this network is:

$$z = v * h(X * U) = \sum_{k=1}^{H} v_k h(X * u_k).$$

The output $z = (z(1), z(2), \ldots, z(T'))$ also has one spatial dimension. Following [156, 251] we derive the distribution of z when $U \sim N(0, \sigma_u^2 \mathbb{I})$ and $v \sim N(0, \sigma_v^2 \mathbb{I})$. The mean is

$$\mathbb{E}[z(t)] = \mathbb{E}\left[\sum_{k=1}^{H} v_k h\left((X * u_k)(t)\right)\right]$$
$$= \mathbb{E}\left[\sum_{k=1}^{H} v_k h\left(\sum_{i=1,j=1}^{c,d} x(i, t+1-j)u_k(i,j)\right)\right].$$

By linearity of expectation and independence of u and v,

$$\mathbb{E}[z(t)] = \sum_{k=1}^{H} \mathbb{E}[v_k]\mathbb{E}\left[h\left((X * u_k)(t)\right)\right] = 0,$$

since v has a mean of zero. The central limit theorem (CLT) can be applied when h is bounded to show that $z(t)$ approaches in distribution to a Gaussian as $H \to \infty$ and σ_v^2 is scaled as $1/H$. Note that u and v don't need to be Gaussian for the CLT to apply, but we will use this property to derive the covariance. This is given by

$$K_z(t_1, t_2) = \mathbb{E}[z(t_1)z(t_2)]$$
$$= \mathbb{E}\left[\sum_{k=1}^{H} v_k^2 h\big((X * u_k)(t_1)\big)h\big((X * u_k)(t_2)\big)\right]$$
$$= H\sigma_v^2 \mathbb{E}\left[h\big((X * u_1)(t_1)\big)h\big((X * u_1)(t_2)\big)\right].$$

The last two steps follow from the independence of u and v and that v is drawn from a zero mean Gaussian. Let $\bar{x}(t) = \text{vec}\left([X(:,t), X(:,t-1), \ldots, X(:,t-d+1)]\right)$ be the flattened tensor with elements within the window of size d at position t of X. Similarly denote $\bar{u} = \text{vec}(u)$. Then the expectation can be written as

$$K_z(t_1, t_2) = H\sigma_v^2 E_u\left[h(\bar{x}(t_1)^T\bar{u})h(\bar{x}(t_2)^T\bar{u})\right]. \tag{5.1}$$

Williams [251] showed $V(x,y) = E_u\left[h(x^Tu)h(y^Tu)\right]$ can be computed analytically for various transfer functions. For example, when $h(x) = \text{erf}(x) = 2/\sqrt{\pi}\int_0^x e^{-s^2}ds$, then

$$V_{\text{erf}}(x,y) = \frac{2}{\pi}\sin^{-1}\frac{x^T\Sigma y}{\sqrt{(x^T\Sigma x)\,(y^T\Sigma y)}}. \tag{5.2}$$

Here $\Sigma = \sigma^2\mathbb{I}$ is the covariance of u. Williams also derived kernels for the Gaussian transfer function $h(x,u) = \exp\{-(x-u)^T(x-u)/2\sigma^2\}$. For the ReLU non-linearity, *i.e.*, $h(t) = \max(0,t)$, Cho and Saul [41] derived the expectation as:

$$V_{\text{relu}}(x, y) = \frac{1}{2\pi}\|x\|\|y\|\big(\sin\theta + (\pi - \theta)\cos\theta\big), \tag{5.3}$$

where $\theta = \cos^{-1}\left(\frac{x^T y}{\|x\|\|y\|}\right)$. We refer the reader to [41, 251] for expressions corresponding to other transfer functions.

Thus, letting σ_v^2 scale as $1/H$ and $H \to \infty$ and for any input X, the output z of our basic convolution-nonlinearity-convolution building block converges to a Gaussian distribution with zero mean and covariance

$$K_z(t_1, t_2) = V\left(\bar{x}(t_1), \bar{x}(t_2)\right). \tag{5.4}$$

5.2.3 Limiting distribution for stationary input

We now consider the case when channels of X are drawn i.i.d. from a stationary distribution. A signal x is stationary (in the weak- or wide-sense) if the mean is position invariant and the covariance is shift-invariant, *i.e.*,

$$m_x = \mathbb{E}[x(t)] = \mathbb{E}[x(t+\tau)], \quad \forall\tau \tag{5.5}$$

and

$$\begin{aligned} K_x(t_1, t_2) &= \mathbb{E}[(x(t_1) - m_x)(x(t_2) - m_x)] \\ &= K_x(t_1 - t_2), \forall t_1, t_2. \end{aligned} \tag{5.6}$$

An example of a stationary distribution is white noise where $x(i)$ is i.i.d. from a zero mean Gaussian distribution $N(0, \sigma^2)$ resulting in a mean $m_x = 0$ and covariance $K_x(t_1, t_2) = \sigma^2 \mathbf{1}[t_1 = t_2]$. Note that the input for the deep image prior is drawn from this distribution.

Theorem 5.1. *Let each channel of X be drawn independently from a zero mean stationary distribution with covariance function K_x. Then the output of a two-layer convolutional network with the sigmoid non-linearity, i.e., $h(t) = \text{erf}(t)$, converges to a zero mean stationary Gaussian process as the number of input channels c and filters H go to infinity sequentially. The stationary covariance K_z is given by*

$$K_z^{\text{erf}}(t_1, t_2) = K_z(r) = \frac{2}{\pi}\sin^{-1}\frac{K_x(r)}{K_x(0)}.$$

where $r = t_2 - t_1$.

The full proof is obtained by applying the continuous mapping thorem [142] on the formula for the sigmoid non-linearity. The theorem implies that the limiting distribution of Z is a stationary GP if the input X is stationary.

Lemma 5.1. *Assume the same conditions as Theorem 5.1 except the non-linearity is replaced by ReLU. Then the output converges to a zero mean stationary Gaussian process with covariance* K_z

$$K_z^{\text{relu}}(t_1, t_2) = \frac{K_x(0)}{2\pi}\Big(\sin\theta^x_{t_1,t_2} + (\pi - \theta^x_{t_1,t_2})\cos\theta^x_{t_1,t_2}\Big), \tag{5.7}$$

where $\theta^x_{t_1,t_2} = \cos^{-1}\left(K_x(t_1,t_2)/K_x(0)\right)$. *In terms of the angles we get the following:*

$$\cos\theta^z_{t_1,t_2} = \frac{1}{\pi}\Big(\sin\theta^x_{t_1,t_2} + (\pi - \theta^x_{t_1,t_2})\cos\theta^x_{t_1,t_2}\Big).$$

This can be proved by applying the recursive formula for ReLU non-linearity [41]. One interesting observation is that, for both non-linearities, the output covariance $K_z(r)$ at a given offset r only depends on the input covariance $K_x(r)$ at the same offset and on $K_x(0)$.

Two or more dimensions. The results of this section hold without modification and essentially the same proofs for inputs with c channels and two or more spatial dimensions by letting t_1, t_2, and $r = t_2 - t_1$ be vectors of indices.

5.2.4 Beyond two layers

So far we have shown that the output of our basic two-layer building block converges to a zero mean stationary Gaussian process as $c \to \infty$ and then $H \to \infty$. Below we discuss the effect of adding more layers to the network.

Convolutional layers. A proof of GP convergence for deep networks was presented in [144], including the case for transfer functions that can be bounded by a *linear envelope*, such as ReLU. In the convolutional setting, this implies that the output converges to GP as the number of filters in each layer simultaneously goes to infinity.

The covariance function can be obtained by recursively applying Theorem 5.1 and Lemma 5.1; stationarity is preserved at each layer.

Bias term. Our analysis holds when a bias term b sampled from a zero-mean Gaussian is added, *i.e.*, $z^{\text{bias}} = z + b$. In this case the GP is still zero-mean but the covariance function becomes $K_z^{\text{bias}}(t_1, t_2) = \sigma_b^2 + K_z(t_1, t_2)$, which is still stationary.

Upsampling and downsampling layers. Convolutional networks have upsampling and downsampling layers to induce hierarchical representations. It is easy to see that downsampling (decimating) the signal preserves stationarity since $K_x^{\downarrow}(t_1, t_2) = K_x(\tau t_1, \tau t_2)$ where τ is the downsampling factor. Downsampling by average pooling also preserves stationarity. The resulting kernel can be obtained by applying a uniform filter corresponding to the size of the pooling window, which results in a stationary signal, followed by downsampling. However, upsampling in general does not preserve stationarity. Therrien [228] describes the conditions under which upsampling a signal with a linear filter maintains stationarity. In particular, the upsampling filter must be band-limited, such as the sinc filter: $\text{sinc}(x) = \sin(x)/x$. If stationarity is preserved the covariance in the next layer is given by $K_x^{\uparrow}(t_1, t_2) = K_x(t_1/\tau, t_2/\tau)$.

Skip connections. Modern convolutional networks have skip connections where outputs from two layers are added $Z = X + Y$ or concatenated $Z = [X; Y]$. In both cases, if X and Y are stationary GPs so is Z. See [69] for a discussion.

5.3 Bayesian inference for deep image prior

5.3.1 Maximum likehoood estimation

Let's revisit the deep image prior for a denoising task. Given an noisy image $\hat{y}$ the deep image prior solves

$$\min_{\theta, x} ||\hat{y} - f(x, \theta)||_2^2,$$

where x is the input and θ are the parameters of an appropriately chosen convolutional network. Both x and θ are initialized randomly from a prior distribution. Optimization is performed using stochastic gradient descent (SGD) over x and θ (optionally x is kept fixed) and relying on early stopping to avoid overfitting (see Figures 5.1 and 5.2). The denoised image is obtained as $y^* = f(x^*, \theta^*)$.

The inference procedure can be interpreted as a maximum likelihood estimate (MLE) under a Gaussian noise model: $\hat{y} = y + \epsilon$, where $\epsilon = N(0, \sigma_n^2 \mathbb{I})$. Bayesian inference suggests we add a suitable prior $p(x, \theta)$ over the parameters and reconstruct the image by *integrating* the posterior to get $y^* = \int p(x, \theta \mid \hat{y}) f(x, \theta) dx d\theta$, The obvious computational challenge is computing this posterior average. An intermediate option is maximum *a posteriori* (MAP) inference where the argmax of the posterior is used. However, both MLE and MAP do not capture parameter uncertainty and can overfit to the data.

5.3.2 Maximum a posterior estimation

In standard MCMC the integral is replaced by a sample average of a Markov chain that converges to the true posterior. However, convergence with MCMC techniques is generally slower than backpropagation for deep networks. Stochastic gradient Langevin dynamics (SGLD) [247] provides a general framework to derive an MCMC sampler from SGD by injecting Gaussian noise into the gradient updates. Let $w = (x, \theta)$. The SGLD update is:

$$\begin{aligned} \Delta_w &= \frac{\epsilon}{2}\Big(\nabla_w \log p(\hat{y} \mid w) + \nabla_w \log p(w)\Big) + \eta_t \\ \eta_t &\sim N(0, \epsilon). \end{aligned} \tag{5.8}$$

where ϵ is the step size. Under suitable conditions, *e.g.*, $\sum \epsilon_t = \infty$ and $\sum \epsilon_t^2 < \infty$ and others, it can be shown that $w_1, w_2, \ldots$ converges to the posterior distribution. The log-prior term is implemented as weight decay.

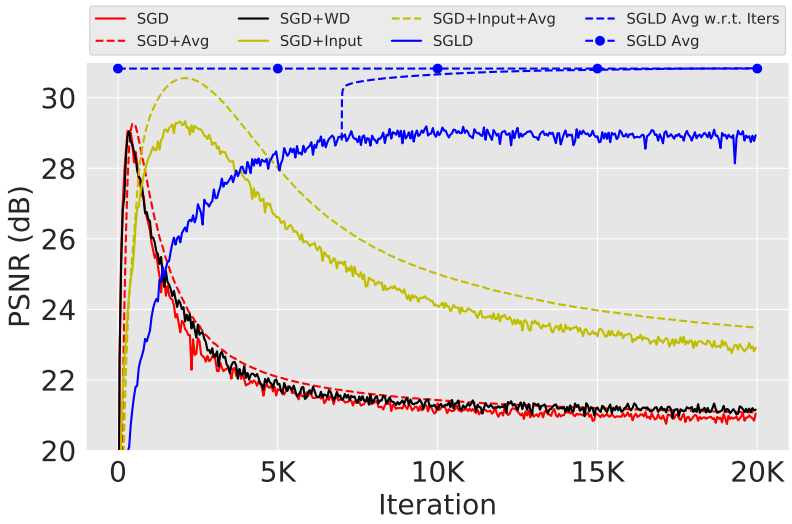


Figure 5.2: The PSNR curve for different learning methods on the "peppers" image of Figure 5.1. The SGD and its variants use early stopping to avoid overfitting. MAP inference by adding a prior term (WD: weight decay) shown as the black curve doesn't avoid overfitting. Moving averages (dashed lines) and adding noise to the input improves performance. By contrast, samples from SGLD after *"burn-in"* remain stable and the posterior mean improves over the highest PSNR of the other approaches.

Our strategy for posterior inference with the deep image prior thus adds Gaussian noise to the gradients at each step to estimate the posterior sample averages after a *"burn in"* phase. As seen in Figure 5.1(a), due to the Gaussian noise in the gradients, the MSE with respect to the noisy image does not go to zero, and converges to a value that is close to the noise level as seen in Figure 5.1(b). It is also important to note that MAP inference alone doesn't avoid overfitting. Figure 5.2 shows a version where weight decay is used to regularize parameters, which also overfits the noise.

5.4 Experiments

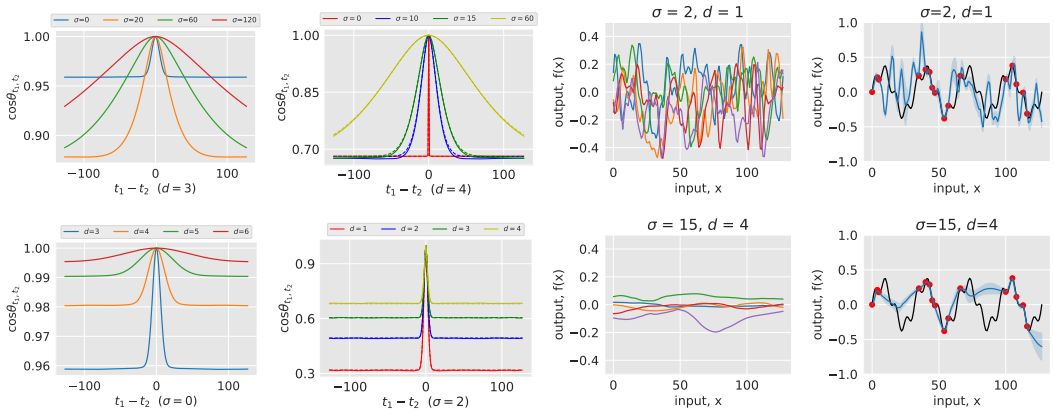
We organized our experiments as follows: Section 5.4.1 illustrates how the input distribution and network architecture influence the stationary GP kernel and illustrate posterior inference using SGLD on a number of toy 1D examples; Section 5.4.3 compares the prior samples and posterior of DIP and its equivalent GP.

5.4.1 Toy examples

We first study the effect of the architecture and input distribution on the covariance function of the stationary GP using 1D convolutional networks. We consider two architectures: (1) AutoEncoder: where d conv + downsampling blocks are followed by d conv + upsampling blocks, and (2) Conv: where convolutional blocks without any upsampling or downsampling. We use ReLU non-linearity after each conv layer in both cases. We also vary the input covariance K_x. Each channel of X is first sampled iid from a zero-mean Gaussian with a variance σ^2. A simple way to obtain inputs with a spatial covariance K_x equal to a Gaussian with standard deviation σ is to then spatially filter channels of X with a Gaussian filter with standard deviation $\sqrt{2}\sigma$.

Figure 5.3 shows the covariance function $\cos\theta_{t_1,t_2} = K_z(t_1 - t_2)/K_z(0)$, induced by varying the σ and depth d of the two architectures (Figure 5.3a-b). We empirically estimated the covariance function by sampling many networks and inputs from the prior distribution. The covariance function for the convolutional-only architecture is also calculated using the recursion in Equation 5.7. For both architectures increasing σ and d introduce longer-range spatial covariances. For the auto-encoder upsampling induces longer-range interactions even when σ is zero shedding some light on the role of upsampling in the deep image prior. Our network architectures have 128 filters, even so, the match between the empirical covariance and the analytic one is quite good as seen in Figure 5.3(b).

Figure 5.3(c) shows samples drawn from the prior of the convolutional-only architecture. Figure 5.3(d) shows the posterior mean and variance with SGLD inference where we randomly dropped 90% of the data from a 1D signal. Changing the covariance influences the mean and variance which is qualitatively similar to choosing the scale of the stationary kernel in the GP: larger scales (bigger input σ or depth) lead to smoother interpolations.



(a) $\cos\theta_{t_1,t_2}$ (AE) (b) $\cos\theta_{t_1,t_2}$ (Conv) (c) Prior (d) Posterior

Figure 5.3: Priors and posterior with 1D convolutional networks. The covariance function $\cos\theta_{t_1,t_2} = K(t_1 - t_2)/K(0)$ for the (a) AutoEncoder and **(b)** Conv architectures estimated empirically for different values of depth and input covariance. For the Conv architecture we also compute the covariance function analytically using recursion in Equation 5.7 shown as dashed lines in panel (b). The empirical estimates were obtained with networks with 256 filters. The agreement is quite good for the small values of Sigma. For larger offsets the convergence towards a Gaussian is approximate. Panel c) shows samples from the prior of the Conv architecture with two different configurations, and panel (d) shows the posterior means and variances estimated using SGLD.

5.4.2 Natural images

Throughout our experiments, we adopt the network architecture reported in [237] for image denoising and inpainting tasks for a direct comparison with their results. These architectures are 5-layer auto-encoders with skip connections and each layer contains 128 channels. We consider images from the standard image reconstruction datasets [49, 97]. For inference, we use a learning rate of 0.01 for image denoising and 0.001 for image inpainting. We compare the following inference schemes:

1. **SGD+Early:** Vanilla SGD with early stopping.
2. **SGD+Early+Avg:** Averaging the predictions with an exponential sliding window of the vanilla SGD.
3. **SGD+Input+Early:** Perturbing the input x with an additive Gaussian noise with mean zero and standard deviation σ_p at each learning step of SGD.

4. **SGD+Input+Early+Avg:** Averaging the predictions of the earlier approach with an exponential window.
5. **SGLD:** Averaging after burn-in iterations of posterior samples with SGLD inference.

We manually set the stopping iteration in the first four schemes to one with essentially the best reconstruction error — note that this is an oracle scheme and cannot be implemented in real reconstruction settings. For the image denoising task, the stopping iteration is set as 500 for the first two schemes and 1800 for the third and fourth methods. For the image inpainting task, this parameter is set as 5000 and 11000 respectively.

The third and fourth variants were described in the supplementary material of [237] and in the released codebase. We found that injecting noise into the input during inference consistently improves results. However, as observed in [237], regardless of the noise variance σ_p, the network is able to drive the objective to zero, it overfits to the noise. This is also illustrated in Figure 5.1 (a-b).

Since the input x can be considered as part of the parameters, adding noise to the input during inference can be thought of as approximate SGLD. It is also not beneficial to optimize x in the objective and is kept constant (though adding noise still helps). SGLD inference includes adding noise to all parameters, x and θ, sampled from a Gaussian distribution with variance scaled as the learning rate η, as described in Equation 5.4. We used 7K burn-in iterations and 20K training iterations for the image denoising task, 20K and 30K for image inpainting tasks. Running SGLD longer doesn't improve results further. The weight-decay hyper-parameter for SGLD is set inversely proportional to the number of pixels in the image and equal to 5e-8 for a 1024×1024 image. For the baseline methods, we did not use weight decay, which, as seen in Figure 5.2, doesn't influence results for SGD.

Image denoising

We first consider the image-denoising task using various inference schemes. Each method is evaluated on a standard dataset for image denoising [49], which consists of 9 colored images corrupted with the noise of $\sigma = 25$.

Figure 5.2 presents the peak signal-to-noise ratio (PSNR) values with respect to the clean image over the optimization iterations. This experiment is on the "peppers" image from the dataset as seen in Figure 5.1. The performance of SGD variants (red, black and yellow curves) reaches a peak but gradually degrades. By contrast, samples using SGLD (blue curves) are stable with respect to PSNR, alleviating the need for early stopping. SGD variants benefit from exponential window averaging (dashed red and yellow lines), which also eventually overfits. Taking the posterior mean after burn-in with SGLD (dashed blue line) consistently achieves better performance. The posterior mean at the 20K iteration (dashed blue line with markers) achieves the best performance among the various inference methods.

Figure 5.4 shows a qualitative comparison of SGD with early stopping to the posterior mean of SGLD, which contains fewer artifacts. Table 5.1 shows the quantitative comparisons between the SGLD and the baselines. We run each method 10 times and report the mean and standard deviations. SGD consistently benefits from perturbing the input signal with noise-based regularization, and from moving averaging. However, as noted, these methods still have to rely on early stopping, which is hard to set in practice. By contrast, SGLD outperforms the baseline methods across all images. Our reported numbers (SGD + Input + Early + Avg) are similar to the single-run results reported in prior work (30.44 PSNR compared to ours of 30.33$\pm$0.03 PSNR.) SGLD improves the average PNSR to 30.81. As a reference, BM3D [49] obtains an average PSNR of 31.68.

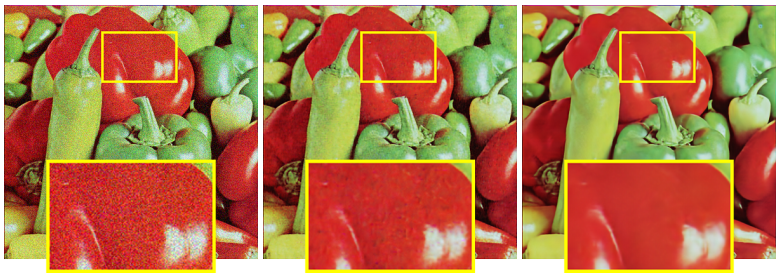


Figure 5.4: Image denoising results. Denoising the input noisy image with SGD and SGLD inference.

Image inpainting

For image inpainting, we experiment on the same task as [237] where 50% of the pixels are randomly dropped. We evaluate various inference schemes on the standard image inpainting dataset [97] consisting of 11 grayscale images.

Table 5.2 presents a comparison between SGLD and the baseline methods. Similar to the image denoising task, the performance of SGD is improved by perturbing the input signal and, additionally by averaging the intermediate samples during optimization. SGLD inference provides additional improvements; it outperforms the baselines and improves over the results reported in [237] from 33.48 to 34.51 PSNR. Figure 5.5 shows qualitative comparisons between SGLD and SGD. The posterior mean of SGLD has fewer artifacts than the best result generated by SGD variants.

Besides gains in performance, SGLD provides estimates of uncertainty. This is visualized in Figure 5.1(d). Observe that uncertainty is low in missing regions that are surrounded by areas of relatively uniform appearance such as the window and floor, and higher in non-uniform areas such as those near the boundaries of different objects in the image.

5.4.3 Equivalence between GP and DIP

We compare the deep image prior (DIP) and its Gaussian process (GP) counterpart, both as prior and for posterior inference, and as a function of the number of

	House	Peppers	Lena	Baboon	F16	Kodak1	Kodak2	Kodak3	Kodak12	Average
SGD + Early	26.74 ±0.41	28.42 ±0.22	29.17 ±0.25	23.50 ±0.27	29.76 ±0.49	26.61 ±0.19	28.68 ±0.18	30.07 ±0.33	29.78 ±0.17	28.08 ±0.09
SGD + Early + Avg	28.78 ±0.35	29.20 ±0.08	30.26 ±0.12	23.82 ±0.11	31.17 ±0.1	27.14 ±0.07	29.88 ±0.12	31.00 ±0.11	30.64 ±0.12	29.10 ±0.05
SGD + Input + Early	28.18 ±0.32	29.21 ±0.11	30.17 ±0.07	22.65 ±0.08	30.57 ±0.09	26.22 ±0.14	30.29 ±0.13	31.31 ±0.08	30.66 ±0.12	28.81 ±0.04
SGD + Input + Early + Avg	30.61 ±0.3	30.46 ±0.03	31.81 ±0.03	23.69 ±0.09	32.66 ±0.06	27.32 ±0.06	31.70 ±0.03	32.86 ±0.08	31.87 ±0.1	30.33 ±0.03
SGLD	**30.86** ±0.61	**30.82** ±0.01	**32.05** ±0.03	**24.54** ±0.04	**32.90** ±0.08	**27.96** ±0.06	**32.05** ±0.05	**33.29** ±0.17	**32.79** ±0.06	**30.81** ±0.08
CMB3D [49]	33.03	31.20	32.27	25.95	32.78	29.13	32.44	34.54	33.76	31.68

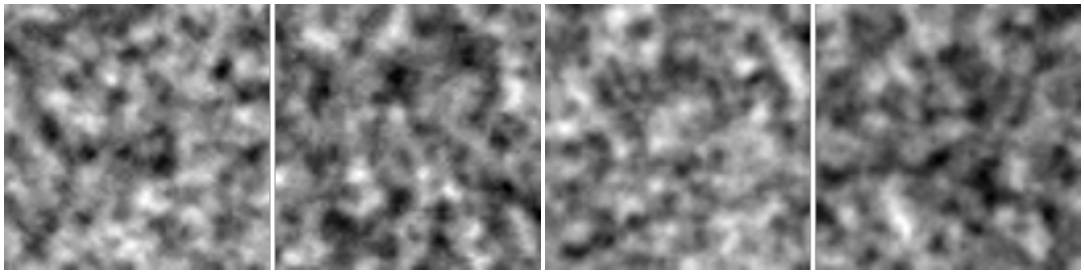
Table 5.1: Image denoising task. Comparison of various inference schemes with the deep image prior for image denoising (σ=25). Bayesian inference with SGLD avoids the need for early stopping while consistently improving results. Details are described in Section 5.4.2.

Method	Barbara	Boat	House	Lena	Peppers	C.man	Couple	Finger	Hill	Man	Montage	Average
SGD + Early	28.48 ±0.99	31.54 ±0.23	35.34 ±0.45	35.00 ±0.25	30.40 ±0.59	27.05 ±0.35	30.55 ±0.19	32.24 ±0.16	31.37 ±0.35	31.32 ±0.29	30.21 ±0.82	31.23 ±0.11
SGD + Early + Avg	28.71 ±0.7	31.64 ±0.28	35.45 ±0.46	35.15 ±0.18	30.48 ±0.6	27.12 ±0.39	30.63 ±0.18	32.39 ±0.12	31.44 ±0.31	31.50 ±0.39	30.25 ±0.82	31.34 ±0.08
SGD + Input + Early	32.48 ±0.48	32.71 ±1.12	36.16 ±2.14	36.91 ±0.19	33.22 ±0.24	29.66 ±0.25	32.40 ±2.07	32.79 ±0.94	33.27 ±0.07	32.59 ±0.14	33.15 ±0.46	33.21 ±0.36
SGD + Input + Early + Avg	33.18 ±0.45	33.61 ±0.3	37.00 ±2.01	37.39 ±0.14	33.53 ±0.31	29.96 ±0.3	33.30 ±0.15	33.17 ±0.77	33.58 ±0.19	32.95 ±0.16	33.80 ±0.6	33.77 ±0.23
SGLD	**33.82** ±0.19	**34.26** ±0.12	**40.13** ±0.16	**37.73** ±0.05	**33.97** ±0.15	**30.33** ±0.15	**33.72** ±0.1	**33.41** ±0.04	**34.03** ±0.03	**33.54** ±0.06	**34.65** ±0.72	**34.51** ±0.08
Ulyanov *et al.* [237]	32.22	33.06	39.16	36.16	33.05	29.80	32.52	32.84	32.77	32.2	34.54	33.48
Papyan *et al.* [169]	28.44	31.44	34.58	35.04	31.11	27.90	31.18	31.34	32.35	31.92	28.05	31.19

Table 5.2: Image inpainting task. Comparison of various inference schemes with the deep image prior for image inpainting. SGLD estimates are more accurate while also providing a sensible estimate of the variance. Details are described in Section 5.4.2.

filters in the network. For efficiency, we used a U-Net architecture with two downsampling and upsampling layers for the DIP.

The above figure shows two samples each drawn from the DIP (with 256 channels per layer) and GP with the *equivalent* kernel. The samples are nearly identical suggesting that the characterization of the DIP as a stationary GP also holds for 2D signals. Next, we compare the DIP and GP on an inpainting task shown in Figure 5.6. The image size here is 64×64. Figure 5.6 top (a) shows the RBF and DIP kernels as a function of the offset. The DIP kernels are heavy-tailed in comparison to Gaussian with support at larger length scales. Figure 5.6 bottom (a) shows the



(a) DIP prior samples (b) GP prior samples

(a) Input (b) SGD (19.23 dB) (c) SGD + Input (19.59 dB) (d) SGLD mean (21.86 dB)

Figure 5.5: Image inpainting using the deep image prior. The posterior mean using SGLD (Panel **(d)**) achieves higher PSNR values and has fewer artifacts than SGD variants.

performance (PSNR) of the DIP as a function of the number of channels from 16 to 512 in each layer of the U-Net, as well as of a GP with the limiting DIP kernel. The PSNR of the DIP approaches the GP as the number of channels increases suggesting that for networks of this size 256 filters are enough for the asymptotic GP behavior. Figure 5.6 (d-e) shows that a GP with the DIP kernel is more effective than one with the RBF kernel, suggesting that the long-tail DIP kernel is better suited for modeling natural images.

While DIPs are asymptotically GPs, the SGD optimization may be preferable because GP inference is expensive for high-resolution images. The memory usage is $O(n^2)$ and running time is $O(n^3)$ for exact inference where n is the number of pixels (*e.g.*, a 500×500 image requires 233 GB memory). The DIP's memory footprint, on the other hand, scales linearly with the number of pixels, and inference with SGD is practical and efficient. This emphasizes the importance of SGLD, which addresses the drawbacks of vanilla SGD and makes the DIP more robust and effective. Finally,

(a) (b) GT (c) Corrupted (d) GP RBF (25.78) (e) GP DIP (26.34) (f) DIP (26.43)

Figure 5.6: Inpainting with a Gaussian process (GP) and deep image prior (DIP). Top (a) Comparison of the Radial basis function (RBF) kernel with the length scale learned on observed pixels in (c) and the stationary DIP kernel. Bottom (a) PSNR of the GP posterior with the DIP kernel and DIP as a function of the number of channels. DIP approaches the GP performance as the number of channels increases from 16 to 512. (d - f) Inpainting results (with the PSNR values) from GP with the RBF (GP RBF) and DIP (GP DIP) kernel, as well as the deep image prior. The DIP kernel is more effective than the RBF.

while we showed that the prior distribution induced by the DIP is asymptotically a GP and the posterior estimated by SGD or SGLD matches the GP posterior for small networks, it remains an open question if the posterior matches the GP posterior for deeper networks.

5.5 Conclusion and subsequent works

We presented a novel Bayesian view of the deep image prior, which parameterizes a natural image as the output of a convolutional network with random parameters and a random input. First, we showed that the output of a random convolutional network converges to a stationary zero-mean GP as the number of channels in each layer goes to infinity, and showed how to calculate the realized covariance. This characterized the deep image prior as approximately a stationary GP. Our work differs from prior work relating GPs and neural networks by analyzing the *spatial* covariance of network activations on a single input image. We then used SGLD to conduct fully Bayesian posterior inference in the deep image prior, which improves performance and prevents the need for early stopping.

Subsequent to our research, Gadelha *et al.* [68] built a *deep manifold prior* for manifold structured data (*e.g.*, surfaces of 3D shapes), demonstrated its effectiveness in a variety of manifold reconstruction applications (*e.g.*, point cloud denoising and interpretation), and characterized its limiting behavior with Gaussian processes.

Historically, using a deep network to parameterize a small number of images was viewed as a limited approach, primarily due to the widespread belief that training a neural network required a substantial volume of data. However, this viewpoint has been significantly challenged, largely driven by a surge in research following the Neural Radiance Fields (NeRF) model [149], which optimizes a fully connected neural network on multiple views to represent a 3D scene. Our work potentially offers valuable insights into the NeRF.

CHAPTER 6

3D GENERATION AND MANIPULATION

Creating and editing the shape and color of 3D objects require tremendous human effort and expertise. In this chapter, we will introduce several machine learning techniques that can automate this task. In particular, Sec. 6.1 introduces a generic multi-modal generative model that couples the 2D modalities and implicit 3D representations through shared latent spaces. With this model, versatile 3D generation and manipulation are enabled by simply propagating the editing from a specific 2D controlling modality through the latent spaces. For example, editing the 3D shape by drawing a sketch, re-colorizing the 3D surface via painting color scribbles on the 2D rendering, or generating 3D shapes of a certain category given one or a few reference images. Sec. 6.2 presents an approach for jointly estimating the camera pose and scene representation from images from a single scene. This approach allows us to operate in the general SE(3) pose setting, unlike the baselines. It works favorably on low-texture and low-resolution images, demonstrating complementary performance to classical Structure-from-Motion (SfM) pipelines.

6.1 Cross-modal 3D shape generation and manipulation

6.1.1 Overview

With the growth in 3D acquisition and visualization technology, there is an increasing need of tools for 3D content creation and editing tasks such as deforming the shape of an object, changing the color of a part, or inserting or removing a component. The graphics and vision community has proposed a number of tools for these

tasks [5, 51, 173, 193]. Yet, manipulating 3D still requires tremendous human labor and expertise, prohibiting wide-scale adoption by non-professionals. Compared to the traditional 3D user interfaces, 2D interactions on view-dependent image planes can be a more intuitive way to edit the shape. This has motivated the community to leverage advances in shape representations using deep networks [37, 147, 170, 205] for 3D shape manipulation with 2D controls, such as mesh reconstruction from sketches [83] and color editing with scribbles [134]. However, most prior works on 2D-to-3D shape manipulation are tailored to a particular editing task and interaction format, which makes generalization to new editing tasks or controls challenging, or even infeasible. This is important because there is often no single interaction that fits every use case – the preferred 2D user control depends on the editing goals, scenarios, devices, or targeted users.

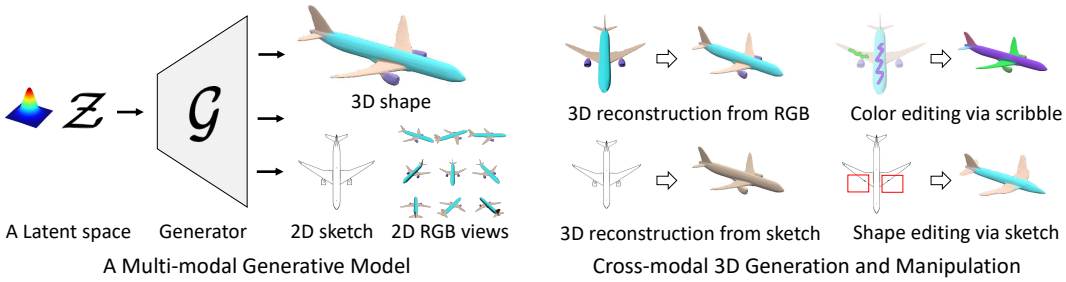


Figure 6.1: Overview. We propose a multi-modal generative model that bridges multiple 2D (*e.g.*, sketch, color views) and 3D modalities via shared latent spaces (*left*). Versatile 3D shape generation and manipulation tasks can be tackled via a simple latent optimization method (*right*).

Motivated by this, we propose a 2D-to-3D framework that not only works on a single control modality but also enjoys the flexibility of handling various types of 2D interactions without the need for changing the architecture or even re-training (Fig. 6.1 left). Our framework bridges various 2D interaction modalities and the target 3D shape through a uniform editing propagation mechanism. The key is to construct a shared latent representation across generative models of each of the 2D

and 3D modalities. The shared latent representation enforces that an arbitrary latent code corresponds to a 3D model that is consistent with every modality, in terms of both shape and color. With our model, any editing can be achieved by an objective that aims to match the corresponding editing modality and backpropagating the error to estimate the latent code. Moreover, different editing operations and modalities can be combined and interleaved leading to a versatile tool for editing the shape (Fig. 6.1 right). The approach can be extended to a new user control by simply adding a generator for the corresponding modality in the framework.

We evaluate our framework on two representative 2D modalities, *i.e.*, grayscale line sketches, and rendered color images. We provide extensive quantitative and qualitative results in shape and color editing with sketches and scribbles, as well as single-view, few-shot, or even partial-view cross-modal shape generation. The proposed method is conceptually simple, easy to implement, robust to input domain shifts, and generalizable to new modalities with no special requirement on the network architecture.

6.1.2 Related works

Methods	*Manipulation*		*Generation*		
	Shape	Color	Single view	Partial view	Few shot
Sketch2Mesh [83]	✓	✗	✓	✗	✗
DualSDF [87]	✓	✗	✗	✗	✗
EditNeRF [134]	✓	✓	✗	✗	✗
Ours	✓	✓	✓	✓	✓

Table 6.1: Comparisons to cross-modal 3D editing and generation works.

Multi-Modal Generative Models. There has been much work on learning a joint distribution of multiple modalities $p(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n)$ where each modality $\boldsymbol{x}_i$ represents one representation (*e.g.*, images, text) of underlying signals. Multi-modal VAEs [113, 201, 215, 254, 255] learn a joint distribution $p_\theta(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n \mid \boldsymbol{z})$ conditioned

on common latent variables $\boldsymbol{z} \in \mathcal{Z}$. Without the assumption of paired multi-modal data, multi-modal GANs [42, 77, 131] learn the joint distribution by sharing a latent space and model parameters across modalities. These multi-modal generative models have enabled versatile applications such as cross-modal image translation [42, 131] and domain adaptation [131]. Similar to these works, we build a multi-modal generative model that bridges multiple modalities via a shared latent space. However, we generate and edit 3D shapes with sparse 2D inputs (*e.g.*, scribbles, sketches) and build a 2D-3D generative model based on variational auto-decoders (VADs) [87, 276]. Prior work [276] has shown that VADs excel at generative modeling from incomplete data. In this work, we demonstrate that the multi-modal VADs (MM-VADs) are ideally suited for the task of 3D generation and manipulation from sparse 2D inputs (*e.g.*, color scribble or partial inputs).

Tab. 6.1 summarizes the commons and differences between our work and recent efforts [83, 87, 134] on 3D manipulation and generation. Similar to Sketch2Mesh [83], we edit and reconstruct 3D shapes from 2D sketches. However, we tackle this problem via a novel multi-modal *generative* model that performs more robustly to input domain shift (*e.g.*, partial input, sparse color scribble). Furthermore, the shape and color edits can be combined and interleaved with our model; Like EditNeRF, we edit the appearance of 3D shapes via 2D color scribbles. However, we conduct the 3D editing via a simple latent optimization, instead of finetuning the network weights per edit; Akin to DualSDF [87], we build a generative model for 3D manipulation, yet we generate and edit shapes from 2D modalities which is more intuitive to edit the shape than using 3D primitives. Moreover, our generative model can be adapted to generate 3D shapes of a certain category (*e.g.*, armchairs) given a few 2D examples, namely, *few-shot cross-modal shape generation*.

6.1.3 Approach

We describe the Variational Auto-Decoders (VADs) [276] in § 6.1.3.1, introduce the proposed VAD-based multi-modal generative model (dubbed MM-VADs) in § 6.1.3.2, and illustrate the application of MM-VADs in cross-modal 3D shape generation and manipulation tasks in § 6.1.3.3.

6.1.3.1 Background: Variational Auto-Decoder

Given observation variables $\boldsymbol{x} \sim p(\boldsymbol{x})$ and latent variables $\boldsymbol{z} \sim p(\boldsymbol{z})$, a variational auto-decoder (VAD) approximates the data distribution $p(\boldsymbol{x})$ via a parametric family of distributions $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ with parameters θ. Similar to variational auto-encoders (VAEs) [113], VADs are trained by maximizing the marginal distribution $p(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$. In practice this integral is expensive or intractable, so the model parameters θ are learned instead by maximizing the Evidence Lower Bound (ELBO):

$$\mathcal{V}(\phi, \theta \mid \boldsymbol{x}) = -\operatorname{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p(\boldsymbol{z})\big) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\big], \tag{6.1}$$

where $\operatorname{KL}(\cdot \parallel \cdot)$ is the Kullback-Leibler divergence that encourages the posterior distribution to follow the latent prior $p(\boldsymbol{z})$, and $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ is an approximation of the posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$. In VAEs, $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ is parametrized by a neural network and ϕ are the parameters of the encoder. In VADs, ϕ are instead learnable similar to the parameters θ in the decoder $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$. For example, the multivariate Gaussian approximate posterior for a data instance $\boldsymbol{x}_i$ is defined as:

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_i) \coloneqq \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{6.2}$$

where $\phi = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$. The reparametrization trick is applied in order to back-propagate the gradients to the mean $\boldsymbol{\mu}_i$ and variance $\boldsymbol{\Sigma}_i$ in VADs. In comparison, VAEs back-propagate the gradients through the mean $\boldsymbol{\mu}_i$ and variance $\boldsymbol{\Sigma}_i$ to learn the parameters of the encoder. At inference time, the parameters ϕ of the approximate posterior distribution can be estimated by maximizing the ELBO in Eqn. 6.1 while the parameters θ of the decoder are frozen:

$$\phi^* = \underset{\phi}{\operatorname{argmax}}\ \mathcal{V}(\phi \mid \theta, \boldsymbol{x}_i). \tag{6.3}$$

Despite the similarity between VAEs and VADs, prior works [276] demonstrate that VADs perform approximate posterior inference more robustly on *incomplete data* and *input domain shifts* than VAEs.

6.1.3.2 Multi-Modal Variational Auto-Decoder

We consider two modalities $\boldsymbol{x}, \boldsymbol{w}$ and an *i.i.d.* dataset with paired instances $(\boldsymbol{X}, \boldsymbol{W}) = \{(\boldsymbol{x}_0, \boldsymbol{w}_0), \ldots, (\boldsymbol{x}_N, \boldsymbol{w}_N)\}$. We target at learning a joint distribution of both modalities $p(\boldsymbol{x}, \boldsymbol{w})$. Like VADs [276], the multi-modal VADs (MM-VADs) are trained by maximizing the ELBO:

$$\mathcal{V}(\phi, \theta \mid \boldsymbol{x}, \boldsymbol{w}) = -\operatorname{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{w}) \parallel p(\boldsymbol{z})\big) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{w})}\big[\log p_\theta(\boldsymbol{x}, \boldsymbol{w} \mid \boldsymbol{z})\big], \tag{6.4}$$

where $\boldsymbol{z}$ is the latent variable shared by the two modalities $\boldsymbol{x}$ and $\boldsymbol{w}$, $p_\theta(\boldsymbol{x}, \boldsymbol{w} \mid \boldsymbol{z}) = p_{\theta_x}(\boldsymbol{x} \mid \boldsymbol{z}) p_{\theta_w}(\boldsymbol{w} \mid \boldsymbol{z})$ under the assumption that the two modalities $\boldsymbol{x}$ and $\boldsymbol{w}$ are independent conditioned on the latent variable $\boldsymbol{z}$ (*i.e.*, $\boldsymbol{x} \perp\!\!\!\perp \boldsymbol{w} \mid \boldsymbol{z}$). In practice, $p_{\theta_x}(\boldsymbol{x} \mid \boldsymbol{z})$ or $p_{\theta_w}(\boldsymbol{w} \mid \boldsymbol{z})$ can be parameterized by different networks for the two modalities $\boldsymbol{x}$ and $\boldsymbol{w}$ respectively. The parameters ϕ of the approximate posterior distribution $q_\phi(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{w})$ are learnable parameters where $\phi = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ under the assumption of multivariate Gaussian posterior distribution. At inference time, the parameters ϕ are estimated via maximizing the ELBO with frozen decoder parameters θ:

$$\phi^* = \underset{\phi}{\operatorname{argmax}}\ \mathcal{V}(\phi \mid \theta, \boldsymbol{x}_i, \boldsymbol{w}_i). \tag{6.5}$$

When one of the modalities is missing during inference, the inputs of the missing modalities are simply set to zero. This is the case when we want to infer one modality from the other (*e.g.*, 3D reconstruction from 2D sketch). This framework can be trivially extended to learn a joint distribution of more than two modalities.

6.1.3.3 Learning a Joint 2D-3D Prior with MM-VADs

Here we introduce the application of MM-VADs in cross-modal 3D shape generation and manipulation. Specifically, we learn a joint distribution of 2D and 3D modalities with MM-VADs. Once trained, MM-VADs can be applied to versatile shape generation and editing tasks via a simple posterior inference (or latent optimization). We explore three representative modalities, including 3D shapes with colorful surfaces, 2D sketches in grayscale, and 2D rendered images in RGB color, donated as $\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{R}$ respectively. Given a dataset $\{(\boldsymbol{C}_i, \boldsymbol{S}_i, \boldsymbol{R}_i)\}$, we target at learning a joint distribution of the three modalities $p(\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{R})$. Fig. 6.2 presents the overview of the MM-VADs framework. We provide more details in the following sections.

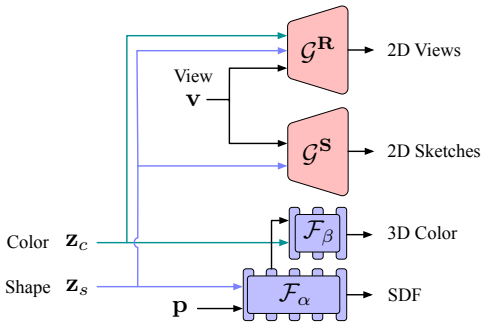


Figure 6.2: Network architecture. We propose a multi-modal variational auto-decoder consisting of a compact shape and color latent space shared across multiple 2D (*e.g.*, sketch, RGB views) or 3D modalities (*e.g.*, signed distance function and 3D surface color).

Joint Latent Space. The MM-VADs share a common latent space $\mathcal{Z}$ across different modalities (Eqn. 6.4). Targeting at editing 3D shape and surface color independently, we further disentangle the shared latent space into the shape and color subspaces, denoted as $\mathcal{Z}_s$ and $\mathcal{Z}_c$ respectively. Therefore, each latent code $\boldsymbol{z} = \boldsymbol{z}_s \oplus \boldsymbol{z}_c$, where $\boldsymbol{z}_s \in \mathcal{Z}_s$, $\boldsymbol{z}_c \in \mathcal{Z}_c$, and $\oplus$ denotes the concatenation operator.

3D Colorful Shape. Targeting at generating and editing 3D shapes and their appearance, we use the 3D colorful shape as one of our modalities. Among various representations of 3D shapes (*e.g.*, voxel, mesh, point clouds), the implicit representations [147, 170, 205] model 3D shapes as isosurfaces of functions and are capable of capturing high-level details. We adopt the DeepSDF [170] to regress the signed

distance functions (SDFs) from point samples directly using an MLP-based *3D shape network* $\mathcal{F}_\alpha(\boldsymbol{z}_s \oplus \boldsymbol{p})$, whose input is a shape latent code $\boldsymbol{z}_s \in \mathcal{Z}_s$ and 3D coordinates $\boldsymbol{p} \in \mathbb{R}^3$. We predict the surface color with another feed-forward *3D color network* $\mathcal{F}_\beta(\boldsymbol{z}_c \oplus \boldsymbol{z}_s^k)$, whose input is a color latent code $\boldsymbol{z}_c \in \mathcal{Z}_c$ and the intermediate features from the k-th layer of 3D shape network $\mathcal{F}_\alpha$. The generator of the 3D modality $\mathcal{G}^{\boldsymbol{C}}$ is the combination of the 3D shape and color network:

$$\mathcal{G}^{\boldsymbol{C}}(\boldsymbol{z}_s \oplus \boldsymbol{z}_c \oplus \boldsymbol{p}) = \big\{\mathcal{F}_\alpha(\boldsymbol{z}_s \oplus \boldsymbol{p}), \mathcal{F}_\beta(\boldsymbol{z}_c \oplus \boldsymbol{z}_s^k)\big\}. \tag{6.6}$$

Both networks are trained using the same set of spatial points. The objective function $\mathcal{L}^{\boldsymbol{C}}$ for $\mathcal{G}^{\boldsymbol{C}}$ is the $\mathcal{L}_1$ loss defined between the prediction and the ground-truth SDF values and surface colors on the sampled points.

2D Sketch. The 2D sketch depicts the 3D structures and provides a natural way for the user to manipulate the 3D shapes. For the purpose of generalization, we adopt a simple and standard fully convolutional network [177] as our sketch generator $\mathcal{G}^{\mathrm{S}}(\boldsymbol{z}_s \oplus \boldsymbol{v})$ with the shape code $\boldsymbol{z}_s \in \mathcal{Z}_s$ and the viewpoint $\boldsymbol{v}$ as input. The objective function $\mathcal{L}^{\boldsymbol{S}}$ is defined as a cross-entropy loss between the reconstructed and ground-truth sketches.

2D Rendering. The 2D color rendering reflects a view-dependent appearance of the 3D surface. Drawing 2D scribbles on the renderings provides an efficient and straightforward interactive tool for the user to edit the 3D surface color. Similar to the 2D sketch modality, we use the standard fully convolutional architecture [177] as our 2D rendering generator $\mathcal{G}^{\boldsymbol{R}}(\boldsymbol{z}_s \oplus \boldsymbol{z}_c \oplus \boldsymbol{v})$, which takes the concatenation of the shape code $\boldsymbol{z}_s \in \mathcal{Z}_s$, the color code $\boldsymbol{z}_c \in \mathcal{Z}_c$ and the viewpoint $\boldsymbol{v}$. We adopt Laplacian-$\mathcal{L}_1$ loss [8] to train $\mathcal{G}^{\boldsymbol{R}}$:

$$\mathcal{L}^{\boldsymbol{R}}(\boldsymbol{z}_i \oplus \boldsymbol{v}, \boldsymbol{R}_i) = \frac{1}{N}\sum_j^J 4^{-j}\big\|\mathrm{L}^j(\mathcal{G}^{\boldsymbol{R}}(\boldsymbol{z}_i \oplus \boldsymbol{v})) - \mathrm{L}^j(\boldsymbol{R}_i)\big\|_1, \tag{6.7}$$

where $\boldsymbol{z}_i$ is the concatenation of the shape and color codes for the target image $\boldsymbol{R}_i$, N is the total number of pixels in the image $\boldsymbol{R}_i$, J is the total number of levels of the

Laplacian pyramid (*e.g.*, 3 by default), and $\mathrm{L}^j(x)$ is the j-th level in the pyramid of image x [25]. This loss encourages sharper output [8] compared to the standard $\mathcal{L}_1$ or MSE loss.

Summary. The proposed MM-VAD framework for learning the joint distribution of the three modalities can be learned with the following objective:

$$\begin{aligned}\mathcal{V}(\phi, \theta \mid \boldsymbol{C}, \boldsymbol{S}, \boldsymbol{R}) = &-\mathrm{KL}\big(q_\phi(\boldsymbol{z} \mid \boldsymbol{C}, \boldsymbol{S}, \boldsymbol{R}) \parallel p(\boldsymbol{z})\big) \\ &+ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{C},\boldsymbol{S},\boldsymbol{R})}\big[\log p_\theta(\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{R} \mid \boldsymbol{z})\big],\end{aligned} \tag{6.8}$$

where the first term regularizes the posterior distribution to a latent prior (*e.g.*, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$), and the second term can be factorized into three components under the assumption that modalities are independent conditioned on the shared latent variable $\boldsymbol{z}$:

$$\begin{aligned}\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{C},\boldsymbol{S},\boldsymbol{R})}\big[\log p_\theta(\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{R} \mid \boldsymbol{z})\big] &= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{C})}\big[\log p_\theta(\boldsymbol{C} \mid \boldsymbol{z})\big] \\ &+ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{C})}\big[\log p_\theta(\boldsymbol{S} \mid \boldsymbol{z})\big] \\ &+ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{C})}\big[\log p_\theta(\boldsymbol{R} \mid \boldsymbol{z})\big] \\ &= \mathcal{L}^{\boldsymbol{C}} + \mathcal{L}^{\boldsymbol{S}} + \mathcal{L}^{\boldsymbol{R}},\end{aligned} \tag{6.9}$$

where each term corresponds to the reconstruction loss per modality as described above. Notice that the 3D shape modality $\boldsymbol{C}$ contains all the information in the latent variable $\boldsymbol{z}$, therefore $q_\phi(\boldsymbol{z} \mid \boldsymbol{C}, \boldsymbol{S}, \boldsymbol{R}) = q_\phi(\boldsymbol{z} \mid \boldsymbol{C})$.

6.1.3.4 Cross-Modal Shape Manipulation with MM-VADs

Given an initial latent code $\boldsymbol{z}_0$ that corresponds to the initial 3D shape $\mathcal{G}^{\boldsymbol{C}}(\boldsymbol{z}_0)$ and any 2D control $\mathcal{G}^{\boldsymbol{M}}(\boldsymbol{z}_0)$ of the 2D modality $\boldsymbol{M} \in \{\boldsymbol{S}, \boldsymbol{R}\}$, the shape manipulation is conducted by optimizing within the latent space to get the updated code $\hat{\boldsymbol{z}}$ such that $\mathcal{G}(\hat{\boldsymbol{z}})^{\boldsymbol{M}}$ matches the 2D edits $\boldsymbol{e}^{\boldsymbol{M}}$:

$$\hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\operatorname{argmin}}\ \mathcal{L}_{\text{edit}}(\mathcal{G}^{\boldsymbol{M}}(\boldsymbol{z}), \boldsymbol{e}^{\boldsymbol{M}}) + \mathcal{L}_{\text{reg}}(\boldsymbol{z}), \tag{6.10}$$

where $\mathcal{L}_{\text{edit}}$ could be any loss (*e.g.*, $\mathcal{L}_1$ loss) that encourages the 2D modalities $\mathcal{G}(\hat{\boldsymbol{z}})^{\boldsymbol{M}}$ to match the 2D edits $\boldsymbol{e}^{\boldsymbol{M}}$, and $\mathcal{L}_{\text{reg}}(\boldsymbol{z})$ encourages the latent code to stay in the latent

prior of MM-VADs. We apply the regularization loss proposed in DualSDF [87]:

$$\mathcal{L}_{\text{reg}} = \gamma \max(\|\boldsymbol{z}\|_2^2, \beta), \tag{6.11}$$

where γ and β control the strength of the regularization loss. The latent optimization is closely related to the posterior inference (Eqn. 6.5) of MM-VADs.

MM-VADs allow free-form edits $e^{\boldsymbol{M}}$. For example, the edits $e^{\boldsymbol{M}}$ could be local modifications on the sketch or sparse color scribbles on 2D renderings. This makes the MM-VADs ideally suited for interactive 3D manipulation tasks. In comparison, the encoder-decoder networks [83] are not robust to the input domain shift (*e.g.*, incomplete data [276]) and require re-training per type of user interactions (*e.g.*, sketch, color scribble).

6.1.3.5 Cross-Modal Shape Generation with MM-VADs

Single-View Reconstruction. Given a single input $\boldsymbol{x}^{\boldsymbol{M}}$ of the 2D modality $\boldsymbol{M} \in \{\boldsymbol{C}, \boldsymbol{R}\}$, the task of single-view cross-modal shape generation is to reconstruct the corresponding 3D shape satisfying the 2D constraint. Without the need of training one model per pair of 2D and 3D modalities [83, 220] or designing differentiable renderers [132] for each 2D modalities [83], like shape manipulation (§6.1.3.4), this task can be tackled via the latent optimization:

$$\hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\text{argmin}} \ \mathcal{L}_{\text{recon}}(\mathcal{G}^{\boldsymbol{M}}(\boldsymbol{z}), \boldsymbol{x}^{\boldsymbol{M}}) + \mathcal{L}_{\text{reg}}(\boldsymbol{z}), \tag{6.12}$$

Partial-View Reconstruction. The MM-VADs are flexible to reconstruct 3D shapes from partially visible inputs. More interestingly, when the input is ambiguous, it provides diverse 3D reconstructions by performing the latent optimization with different initialization of the latent code $\boldsymbol{z}$. This property has practical applications. For example, the MM-VAD could provide multiple 3D shape suggestions interactively while the user is drawing sketches.

Few-Shot Generation. Given a few 2D images spanning a subspace in the 3D distribution that represents a certain semantic attribute (*e.g.*, armchairs, red chairs),

the task of few-shot shape generation is to learn a 3D shape generative model that conceptually aligns with the provided 2D images. Given our pre-trained MM-VAD, we tackle this task by steering the latent space with adversarial loss, borrowing the idea from MineGAN [245]. Specifically, we learn a mapping function $h_\omega(\boldsymbol{z})$ that maps the prior distribution of the latent space $\boldsymbol{z} \sim \hat{p}(\boldsymbol{z})$ (*i.e.*, $\mathcal{N}(\mathbf{0}, \mathbf{I})$) to a new distribution such that samples from the 2D generators $\mathcal{G}^{\boldsymbol{M}}(h_\omega(\boldsymbol{z}))$ aligns the target data distribution $\boldsymbol{x} \sim \hat{p}(\boldsymbol{x})$ depicted by the provided 2D images. We apply the WGAN-GP loss [84] with frozen generators to learn the mapping function $h_\omega(\boldsymbol{z})$:

$$\min_{\omega} \max_{\mathcal{D}} \; \mathbb{E}_{\boldsymbol{x} \sim \hat{p}(\boldsymbol{x})}\big[\mathcal{D}(\boldsymbol{x})\big] - \; \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}\big[\mathcal{D}(\mathcal{G}^{\boldsymbol{M}}(h_\omega(\boldsymbol{z})))\big], \tag{6.13}$$

where both the mapping function h_ω and the discriminator $\mathcal{D}$ are trained from scratch.

6.1.4 Experiments

This section provides qualitative and quantitative results of the proposed MM-VADs in versatile tasks of 3D shape manipulation (§ 6.1.4.1) and generation (§ 6.1.4.2).

Dataset. We conduct evaluations and comparisons mainly on 3D ShapeNet dataset [28]. For 3D shapes, We follow DeepSDF [170] to sample 3D points and their signed distances to the object surface. The points that are far from the surface (*i.e.*, with an absolute distance higher than a threshold) are assigned a pre-defined background color (*e.g.*, white) while points surrounding the surface are assigned the color of the nearest surface point. For 2D sketches, we use suggestive contours [50] to generate the synthetic sketches. For 2D renderings, we randomize the surface color of 3D shapes per semantic part. We use ShapeNet chairs and airplanes with the same training and test splits as DeepSDF [170].

Implementation Details. We use an 8-layer MLP as the 3D shape network which outputs SDF and a 3-layer MLP as the 3D color network which predicts RGB. We concatenate the features from the 6-th hidden layer of the 3D shape network with

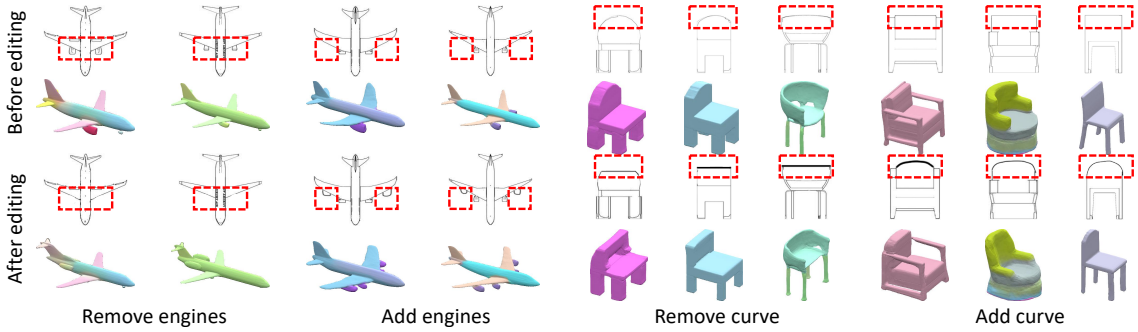


Figure 6.3: Editing shape via sketch. The proposed method enables fine-grained editing of shape geometry, *e.g.*, removing the engine of an airplane or reshaping the back of a chair. Interestingly, new engines often appear at the tail of an airplane after removing the engines on the wing. This is because airplanes without any engines rarely exist in the domain of our generative model. The edited local regions are highlighted in red bounding boxes.

the color code as the input to the 3D color network. We train our MM-VADs using Adam [112].

Baselines. We use the following state-of-the-arts as our baselines:

- **Encoder-Decoder Networks** [83]. This model is trained per task of 3D generation from 2D modalities (sketches or RGB images). We do not use the differentiable rendering proposed in [83] which requires auxiliary information (*e.g.*, segmentation mask, depth) and is applicable to MM-VADs.

- **EditNeRF** [134]. This model edits 3D neural radiance field (including shape and color) by updating the neural network weights based on the user's scribbles. We make comparisons with the pre-trained EditNeRF models.

6.1.4.1 Cross-modal Shape Manipulation

Sketch-Based Shape Manipulation. The proposed MM-VADs allow users to edit the fine geometric structures via 2D sketches, as described in § 6.1.3.4. We provide users with an interactive interface where users can edit the initial sketch by adding or removing a certain part or even deforming a contour line. Fig. 6.3 presents some

	Airplane		Chair	
	− engine	+ engine	− curve	+ curve
Initial shape	0.096	**0.123**	0.066	**0.085**
Edited shape	**0.059**	0.134	**0.054**	0.124

Table 6.2: Editing shape via sketch. We report the Chamfer distance (CD) between the manually edited shapes and our editing results (*lower* is *better*).

qualitative results of sketch-based shape manipulation. Interestingly, we find that our manipulation is semantics-aware. For example, removing the airplane engines on the wings will automatically add new engines to the tail. Such shape priors are absent in non-generative models (*e.g.*, EditNeRF [134]).

It is challenging to quantitatively evaluate sketch-based shape editing due to the lack of ground-truth paired 3D shapes before and after editing. For this reason, prior works [83] report the quantitative results of 3D reconstruction from sketches as a proxy. We follow prior works and report the same quantitative evaluations in Sec. 6.1.4.2. Furthermore, we manually edit the 3D shapes presented in Fig. 6.3 such that their sketches align with the human edits. Tab. 6.2 reports the Chamfer distance (CD) between the manually edited shapes and our editing results. We see that CD improves when removing a part, but adding parts unfortunately increases the CD as it induces more changes to the overall shape. This is often desirable, but the CD metric does not reflect that.

Fig. 6.4 provides a comparison with DualSDF [87]. A fair comparison is not possible, as DualSDF edits shape via 3D primitives instead of 2D views. We find that DualSDF requires users to select *right* primitives to achieve certain edits (*e.g.*, adding a curve to the chair back). In comparison, our sketch-based shape editing is more intuitive.

Scribble-Based Color Manipulation. MM-VADs allow users to edit the appearance of 3D shapes via color scribbles. Fig. 6.5 shows that MM-VADs propagate the

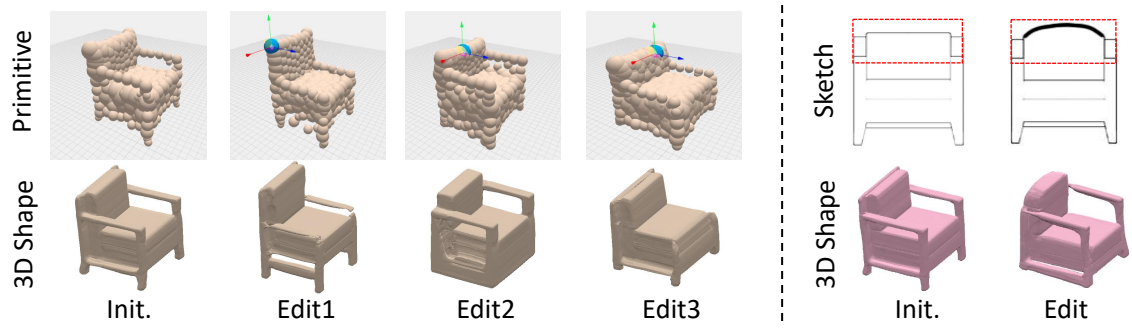


Figure 6.4: Comparison with DualSDF. **Left:** DualSDF [87] edits 3D shapes via 3D primitives. Editing different primitives on the same part may lead to dramatically different editing results (2nd - 4th columns). **Right:** our sketch-based interactions is more intuitive for the user.

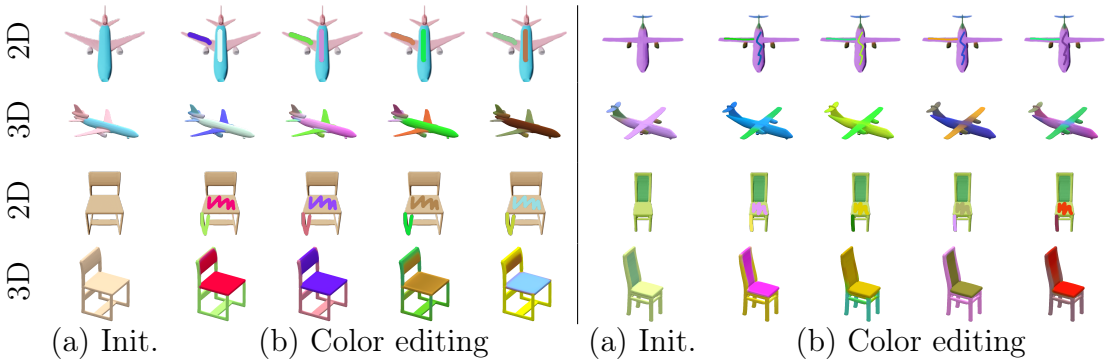


Figure 6.5: Editing shape via color scribble. (a) presents the initial 2D and 3D view of the object. **(b)** shows the 2D color scribbles and 3D color editing results.

sparse color scribbles into desired regions (*e.g.*, from the left wing of the airplanes to the right, from the left leg of chairs to the right). As a quantitative evaluation, we select 10 shapes per category (including airplanes and chairs) and edit the surface color to make it visually similar to reference shapes with the same geometry yet different surface color. The editing quality is measured by the similarity between the renderings of the edited 3D shapes and the reference shapes. Tab. 6.3 reports the PSNR and LPIPS [280] metrics of the evaluation. The surface color of 3D shapes is much closer to the reference after editing, compared to the initial shapes, suggesting the effectiveness of our MM-VAD model in editing color via scribbles.

A similar task has recently been explored in EditNeRF [134]. However, an apple-to-apple comparison with EditNeRF is not possible due to the intrinsically different 3D representations (NeRF [149] vs SDFs [170]). Moreover, the proposed MM-VADs are generative models while EditNeRF is non-generative; The MM-VADs bridge 2D and 3D via shared latent spaces while EditNeRF relies on differentiable rendering. We provide qualitative comparisons with EditNeRF on chairs with similar structures using their pre-trained models. Fig. 6.6 shows that the color editing from MM-VADs is on par with EditNeRF. The MM-VADs achieve the editing via simple latent optimization (Eqn. 6.12), while EditNeRF requires updating the network weights per instance and fails to generate meaningful color editing results via optimizing the color code alone. Furthermore, MM-VADs take 0.06 seconds per edit and 6.78 seconds to render our 3D shapes into 256×256 RGB images, while EditNeRF takes over a minute per edit including rendering.

Methods	Airplane		Chair	
	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
Initial	19.84	0.23	16.20	0.33
Edited	**26.41**	**0.13**	**22.08**	**0.20**

Table 6.3: Quantitative results of editing 3D via 2D scribbles. We edit the surface color of 3D shape based on reference shapes, and report the similarity between the editing results and the target (bottom row). As a reference, we also report the metrics before editing (top row).

6.1.4.2 Cross-Modal Shape Generation

Single-View and Partial-View Shape Reconstruction. Fig. 6.7 compares the performance of our model and the encoder-decoder networks [83] under different occlusion ratios in the lower part of the objects in 2D views. The proposed model only has a slight performance drop as the occluded parts increase (Fig. 6.7a), mainly because of the ambiguity of 3D reconstruction given partial views. In fact, our reconstructions

Figure 6.6: Comparison with EditNeRF. Our model (**bottom**) achieves comparable editing performance with EditNeRF [134] (**top**). We provide three color edits on 2D views (**odd columns**), each followed by the 3D editing result (**even columns**).

results fit the partial views quite well. Even though our model performs slightly worse than the encoder-decoder networks on full-view inputs, the proposed model is more robust to the input domain shift. This is because compared to task-specific training, our model achieves a better trade-off between reconstruction accuracy and domain generalization. More interestingly, our model can achieve diverse and reasonable 3D reconstruction by sampling different initialization for latent optimization (Fig. 6.7b).

Few-Shot Shape Generation. The proposed method is able to adapt the pre-trained multi-modal generative model with as few as 10 training samples of a specific 2D modality. Fig. 6.8 presents some of the few-shot cross-modal shape generation results. To quantitatively evaluate the few-shot shape generation performance, we render the 3D shapes into 2D RGB images and report the Frechet Inception Distance (FID) scores [99] between the rendered images and the ground-truth samples. Since the FID score is not sensitive to the semantic difference between two image sets, we also report the classification error on the random samples from the model before and after the adaptation. Specifically, we train a binary image classifier to identify the target image categories (*e.g.*, armchairs vs. other chairs), and we run the trained classifier on the 2D renderings of the 3D samples before and after the adaptation. As presented in Tab. 6.4, our pre-trained generative model can be effectively adapted to a certain shape subspace given as few as 10 2D examples. This capability allows us to agilely adapt our generative model to a subspace defined by a few unlabelled samples,

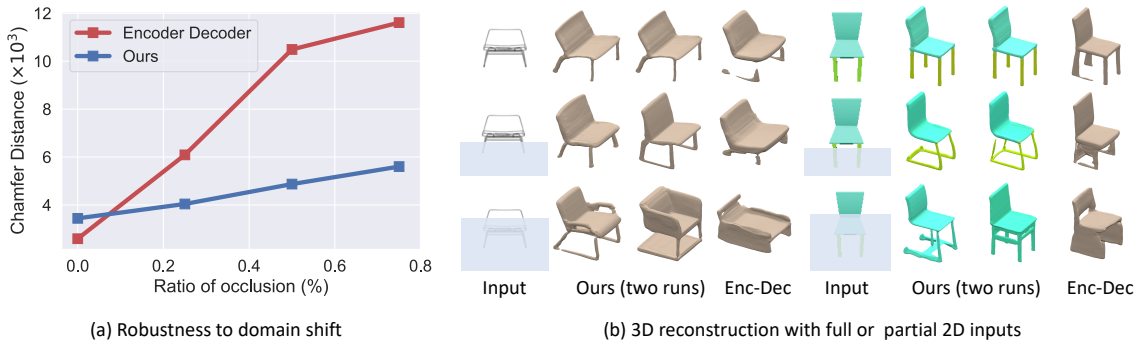


Figure 6.7: 3D reconstruction. (a) Robustness to domain shift. We report the Chamfer distance (*lower* is *better*) between 3D reconstructions and the groundtruth under different ratios of image occlusion. **(b) 3D reconstruction with full or partial 2D inputs.** When the full views are available, our model produces consistent 3D reconstruction in different trials. When only partial views are given, our model produces multiple different 3D reconstructions. In comparison, the encoder-decoder networks [83] trained on full-view sketches are not robust to the domain shift induced by occlusion and unable to provide multiple 3D shapes given partial views. Notice that the predictions of surface color is not available in the encoder-decoder networks from the prior work [83].

so that users can easily narrow down the target shape during the manipulation by providing a few samples of a common attribute, such as a specific category, style, or color. We are unaware of any prior works that can tackle this task in the literature.

Shape and Color Transfer. Transferring shape and color across different 3D instances can be achieved by simply swapping the latent codes. Fig. 6.9 shows that the shape and color are well disentangled in the proposed generative model. The transfer results also are semantically meaningful, *i.e.*, the color is only transferred across the same semantic parts (*e.g.*, seats for the chair, wings for the airplane) even though the geometry of the source and target instances are quite different.

6.1.4.3 Case Study on Real Images

The workflow of 3D designers usually starts by drawing a 2D sketch to portray the coarse 3D geometry and then colorizes the sketch to depict the 3D appearance. These 2D arts are used as a reference to build 3D objects. Undoubtedly this procedure

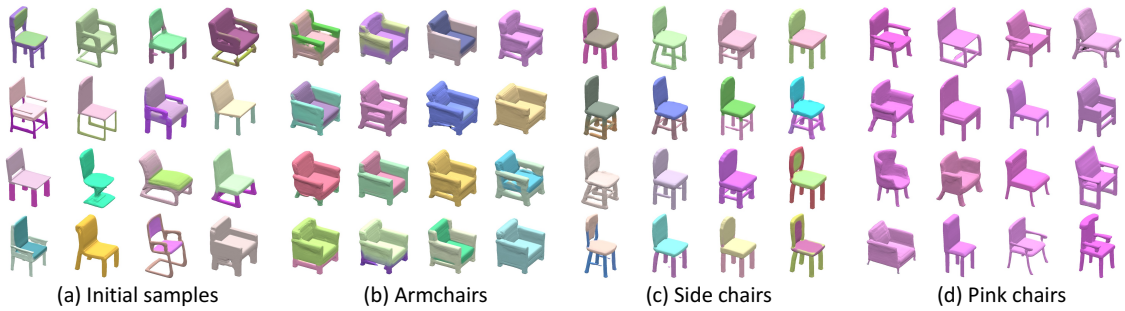


Figure 6.8: Few-shot cross-modal shape generation. (a) presents random 3D samples from our model before the adaptation. Given a few 2D exemplars of a certain category (*e.g.*, armchair), our model can be adapted to generate corresponding 3D shapes (b-d).

Stage	Metrics	Arm	Side	Red	Avg.
Init.	FID ↓	138.1	95.2	93.7	109.0
	Cls.Err. ↓	0.79	0.64	0.82	0.75
Adapt.	FID ↓	**130.4**	**92.4**	**93.0**	**105.3**
	Cls.Err. ↓	**0.01**	**0.10**	**0.00**	**0.04**

Table 6.4: Quantitative results of few-shot cross-modal shape generation. We report Frechet Inception Distance (FID) (*lower* is *better*) and classification error (Cls. Err) (*lower* is *better*). We effectively adapt the pretrained multi-modal VAD model using a few 2D images to a desired 3D shape generator. As a reference, we report the metrics before the few-shot adaptation (top row).

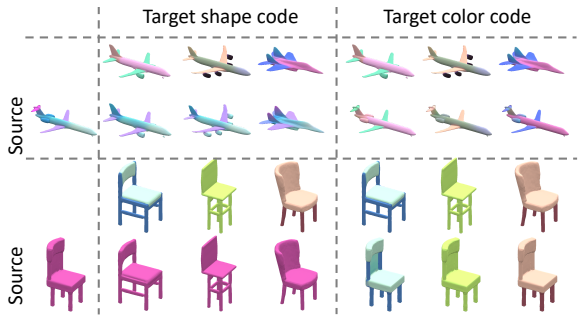


Figure 6.9: Shape and color transfer. The reference 3D shapes (top row) provide the shape codes or color codes for each source instances (first column).

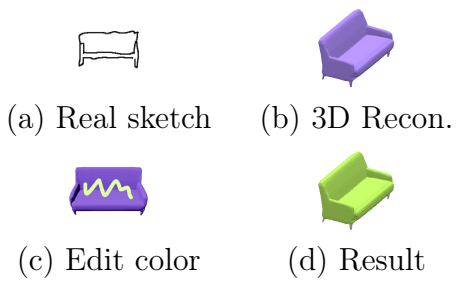


Figure 6.10: Our model enables consecutive 3D reconstruction and manipulation given a hand-drawn sketch.

requires extensive human efforts and expertise. Such tasks can be automated with our MM-VADs. As shown in Fig. 6.10, we first reconstruct the 3D shape from a hand-drawn sketch. We then assign a surface color by randomly sampling a color code from the latent space of the MM-VADs, which can be easily edited by drawing color scribbles on the surface. Our model does not require any re-training on each of these steps and provides a tool to conduct shape generation and color editing consecutively. Such a task is infeasible with the existing works that train an encoder-decoder network to predict 3D shape from sketch [83].

6.1.5 Conclusion

We propose a multi-modal generative model which bridges multiple 2D and 3D modalities through a shared latent space. One limitation of the proposed method is that we are only able to provide editing results in the prior distribution of our generative model. Despite this limitation, our model has enabled versatile cross-modal 3D generation and manipulation tasks without the need of re-training per task and demonstrates strong robustness to input domain shift.

6.2 Joint estimation of 3D scene and camera poses

6.2.1 Overview

NeRF [149] was introduced as a powerful method to tackle the problem of learning neural scene representations and photorealistic view synthesis, and subsequent research has focused on addressing its limitations to extend its applicability to a wider range of use cases (see [221, 267] for surveys). One of the few remaining hurdles for view synthesis in the wild is the need for accurate localization. As images captured in the wild have unknown poses, these approaches often use Structure-from-Motion (SfM) [167, 196] to determine the camera poses. There is often no recourse

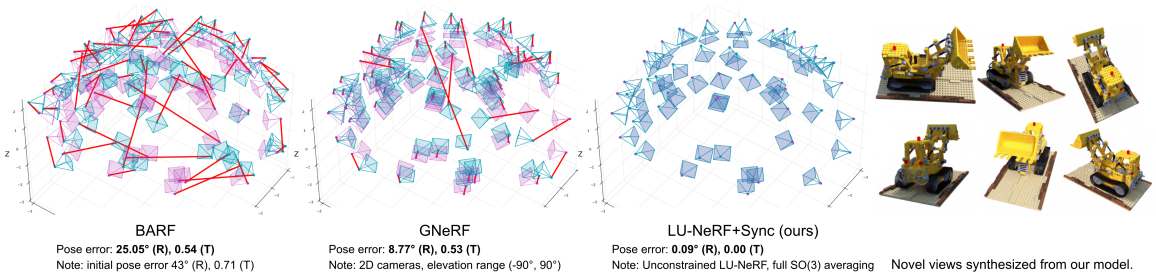


Figure 6.11: Jointly optimizing camera poses and scene representation over a full scene is difficult and under-constrained. This example is the Lego scene with 100 images from the Blender dataset. **Left**: When provided noisy observations of the true camera locations, BARF [126] cannot converge to the correct poses. **Middle**: GNeRF [145] assumes a 2D camera representation (azimuth, elevation) which is accurate for the Blender dataset which has that exact configuration (upright cameras on a sphere). However, GNeRF also requires an accurate prior distribution on poses for sampling. The Lego images live on one hemisphere, but when GNeRF's prior distribution is the full sphere it also fails to localize the images accurately. **Right**: Our full model, LU-NeRF+Sync, is able to recover poses almost perfectly in this particular example. By taking a local-to-global approach, we avoid having strong assumptions about camera representation or pose priors. Following [126, 145] pose errors for each method are reported after optimal global alignment of estimated poses to ground truth poses. To put the translation errors in context, the Blender cameras are on a sphere of radius 4.03.

when SfM fails (see Fig. 6.17 for an example), and in fact, even small inaccuracies in camera pose estimation can have a dramatic impact on photorealism.

Few prior attempts have been made to reduce the reliance on SfM by integrating pose estimation directly within the NeRF framework. However, the problem is severely underconstrained (see Fig. 6.11) and current approaches make additional assumptions to make the problem tractable. For example, NeRf−− [246] focuses on pose estimation in forward-facing configurations, BARF [126] initialization must be close to the true poses, and GNeRF [145] assumes a 2D camera model (upright cameras on a hemisphere).

We propose an approach for jointly estimating the camera pose and scene representation from images from a single scene while allowing for a more general camera configuration than previously possible. Conceptually, our approach is organized in

a local-to-global learning framework using NeRFs. In the *local* processing stage we partition the scene into overlapping subsets, each containing only a few images (we call these subsets *mini-scenes*). Knowing images in a mini-scene are mostly nearby is what makes the joint estimation of pose and scene better conditioned than performing the same task globally. In the *global* stage, the overlapping mini-scenes are registered in a common reference frame through pose synchronization, followed by jointly refining all poses and learning the global scene representation.

This organization into mini-scenes requires learning from a few local unposed images. Although methods exist for few-shot novel view synthesis [30, 31, 54, 117, 158, 275], and separately for optimizing unknown poses [126, 145, 246], the combined setting presents new challenges. Our model must reconcile the ambiguities prevalent in the local unposed setting – in particular the mirror symmetry ambiguity [165], where two distinct 3D scenes and camera configurations produce similar images under affine projection.

We introduce a Local Unposed NeRF (LU-NeRF) model to address these challenges in a principled way. The information from the LU-NeRFs (estimated poses, confidences, and mirror symmetry analysis) is used to register all cameras in a common reference frame through pose synchronization [52, 89, 181], after which we refine the poses and optimize the neural scene representations using all images. In summary, our key contributions are:

- A local-to-global pipeline that learns both the camera poses in a general configuration and a neural scene representation from only an unposed image set.
- LU-NeRF, a novel model for few-shot local unposed NeRF. LU-NeRF is tailored to the unique challenges we have identified in this setting, such as reconciling mirror-symmetric configurations.

Each phase along our local-to-global process is designed with robustness in mind, and the consequence is that our pipeline can be successful even when the initial mini-scenes contain frequent outliers (see Sec 6.2.4 for a discussion on different mini-scene construction techniques). The performance of our method surpasses prior works that jointly optimize camera poses and scene representation, while also being flexible enough to operate in the general SE(3) pose setting unlike prior techniques. Our experiments indicate that our pipeline is complementary to the feature-based SfM pipelines used to initialize NeRF models, and is more reliable in low-texture or low-resolution settings.

6.2.2 Related works

Neural scene representation with unknown poses. BARF [126] and GARF [40] jointly optimize neural scene and camera poses, but require good initialization (*e.g.*, within 15° of the groundtruth). NeRF−− [246], X-NeRF [176], SiNeRF [258], and SaNeRF [33] only work on forward-facing scenes; SAMURAI [22] aims to handle coarsely specified poses (octant on a sphere) using a pose multiplexing strategy during training; GNeRF [145] and VMRF [278] are closest to our problem setting. They do not require accurate initialization and work on 360° scenes. However, they make strong assumptions about the pose distribution, assuming 2DoF and a limited elevation range. Performance degrades when the constraints are relaxed.

Approaches that combine visual SLAM with neural scene representations [182, 212, 292] typically rely on RGB-D streams and are exclusively designed for video sequences. The use of depth data significantly simplifies both scene and pose estimation processes. There are several parallel efforts to ours in this field. For instance, NoPe-NeRF [19] trains a NeRF without depending on pose priors; however, it relies on monocular depth priors. In a manner akin to our approach, LocalRF [148] progressively refines camera poses and radiance fields within local scenes. Despite

this similarity, it presumes monocular depth and optical flow as supervision, and its application is limited to ordered image collections; MELON [123] optimizes NeRF with unposed images using equivalence class estimation, yet it is limited to $\mathbf{SO}(3)$; RUST [188] and FlowCam [206] learn a generalizable neural scene representation from unposed videos.

In summary, prior work on neural scene representation with unknown poses assumes either small perturbations [40, 126, 246, 258], a narrow distribution of camera poses [145, 278], or depth priors [19, 148]. To the best of our knowledge, we are the first to address the problem of neural rendering with unconstrained unknown poses for both ordered and unordered image collections.

Few-shot scene estimation. Learning scene representations from a few images has been studied in [30, 31, 54, 117, 158, 275]. PixelNeRF [275] uses deep CNN features to construct NeRFs from few or even a single image. MVSNeRF [30] leverages cost-volumes typically applied in multi-view stereo for the same task, while DS-NeRF [54] assumes depth supervision is available to enable training with fewer views. Our approach to handle the few-shot case relies on a standard neural field optimization with strong regularization, similar to RegNeRF [158].

Unsupervised pose estimation. There are a number of techniques that can learn to predict object pose from categorized image collections without explicit pose supervision. Multiple views of the same object instance are used in [104, 235] to predict the shape and pose while training is self-supervised through shape rendering. RotationNet [110] uses multiple views of an object instance to predict both poses and class labels but is limited to a small set of discrete uniformly spaced camera viewpoints. The multi-view input is relaxed in [151, 256] which operates on single image collections for a single category. UNICORN [151] learns a disentangled representation that includes pose and utilizes cross-instance consistency at training, while an assumption about object symmetry guides the training in [256].

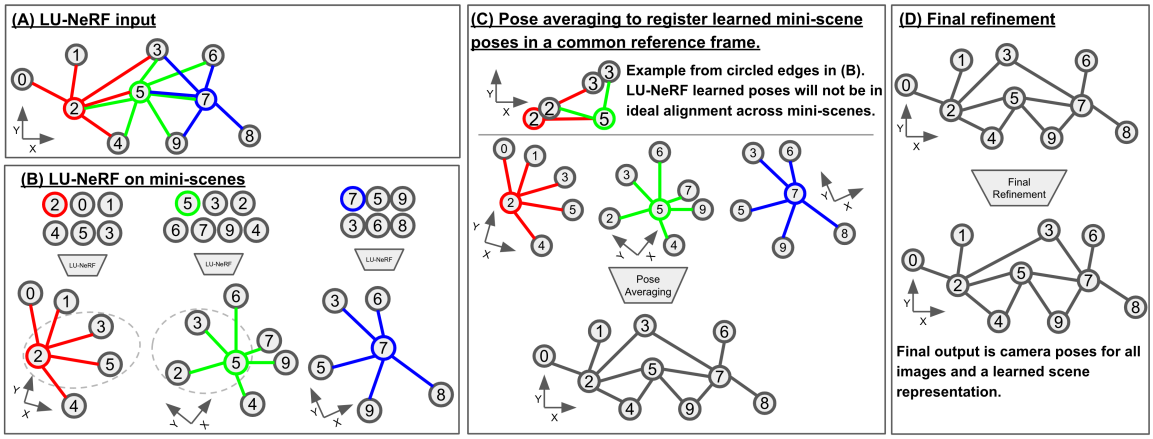


Figure 6.12: Proposed method. (A) shows the ground truth locations of each image (we show this only for visualization). Edge colors show the grouping within mini-scenes. We create a mini-scene for each image, though here only three mini-scenes are highlighted; the ones centered at image 2 (red edges), image 5 (green edges), and image 7 (blue edges). Depending on the strategy used to create mini-scenes, the grouped images can contain outlier images far from the others. (B) LU-NeRF takes unposed images from a single mini-scene and optimizes poses without any constraints on the pose representation. (C) The reference frame and scene scale learned by LU-NeRF is unique to each mini-scene. This, plus estimation errors, means the relative poses between images in overlapping mini-scenes will not perfectly agree. To register the cameras in a common reference frame, we utilize pose synchronization which seeks a globally optimal positioning of all cameras from noisy relative pose measurements – this is possible since we have multiple relative pose estimations for many pairs of images. (D) Lastly, we jointly refine the synchronized camera poses and learn a scene representation.

6.2.3 Approach

An illustration of our approach is shown in Figure 6.12. At the core of our method is the idea of breaking up a large scene into mini-scenes to overcome the non-convexity of global pose optimization without accurate initialization. When the camera poses in the mini-scene are close to one another, we are able to initialize the optimization with all poses close to the identity and optimize for relative poses. In Sec. 6.2.4, we describe how we construct mini-scenes, and below we describe the process of local shape estimation followed by global synchronization.

6.2.3.1 Local pose estimation

The local pose estimation step takes in mini-scenes of typically three to five images and returns the relative poses between the images. The model, denoted LU-NeRF-1, is a small NeRF [149] that jointly optimizes the camera poses as extra parameters as in BARF [126]. In contrast with BARF, in this stage, we are only interested in a rough pose estimation that will be improved upon later, so we aim for a lightweight model with faster convergence by using small MLPs and eliminating positional encoding and view dependency. As we only need to recover relative poses, without loss of generality, we freeze one of the poses at identity and optimize all the others.

Few-shot radiance field optimization is notoriously difficult and requires strong regularization [158]. Besides the photometric ℓ_2 loss proposed in NeRF, we found that adding a loss term for the total variation of the predicted depths over small patches is crucial for the convergence of both camera pose and scene representation:

$$\frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}}\sum_{i,j=1}^{K}\big(d_\theta(\mathbf{r}_{i,j})-d_\theta(\mathbf{r}_{i,j+1})\big)^2+\big(d_\theta(\mathbf{r}_{i,j})-d_\theta(\mathbf{r}_{i+1,j})\big)^2$$

where $\mathcal{R}$ is a set of ray samples, $d_\theta(\mathbf{r})$ is the depth rendering function for a ray $\mathbf{r}$, θ are the model parameters and camera poses, K is the patch size, and (i, j) is the pixel index.

6.2.3.2 Mirror-symmetry ambiguity

The ambiguities and degeneracies encountered when estimating 3D structure have been extensively studied [16, 44, 217]. One particularly relevant failure mode of SfM is distant small objects, where the perspective effects are small and can be approximated by an affine transform, and one cannot differentiate between reflections of the object around planes parallel to the image plane [165]. When enforcing multi-view consistency, this effect, known as mirror-symmetry ambiguity, can result in two different configurations of structure and motion that cannot be told apart (see

Fig. 6.13). We notice, perhaps for the first time, that neural radiance fields with unknown poses can degenerate in the same way.

One potential solution to this problem would be to keep the two possible solutions and drop one of them when new observations arrive. This is not applicable to our case since at this stage the only information available is the few images of the mini-scene.

To mitigate the issue, we introduce a second stage for the training, denoted LU-NeRF-2. We take the estimated poses in world-to-camera frame $\{R_i\}$ from LU-NeRF-1, and the reflected cameras $\{R_\pi R_i\}$, where R_π is a rotation around the optical axis. Note that this is different than post-multiplying by R_π, which would correspond to a global rotation that wouldn't change the relative poses that we are interested in at this stage. We then train two new models, with the scene representation started from scratch and poses initialized as the original and reflected sets, and resolve the ambiguity by picking the one with the smallest photometric training loss. The rationale is that while the issue is caused by LU-NeRF-1 ignoring small perspective distortions, the distortions can be captured on the second round of training, which is easier since one of the initial sets of poses is expected to be reasonable.

6.2.3.3 Local to global pose estimation

After training LU-NeRF-2, we have sets of relative poses for each mini-scene in some local frame. The problem of finding a global alignment given a set of noisy relative poses is known as pose synchronization or pose averaging. It is formalized as optimizing the set of N global poses $\{P_i\}$ given relative pose observations R_{ij},

$$\underset{P \in \mathbf{SE}(3)^N}{\operatorname{argmin}} \; d(P_{ij}, P_j P_i^\top), \tag{6.14}$$

for some metric $d\colon \mathbf{SE}(3) \times \mathbf{SE}(3) \mapsto \mathbb{R}$. The problem is challenging due to non-convexity and is an active subject of research [7, 52, 181]. We use the Shonan rotation method [52] to estimate the camera rotations, followed by a least-squares optimization of the translations.

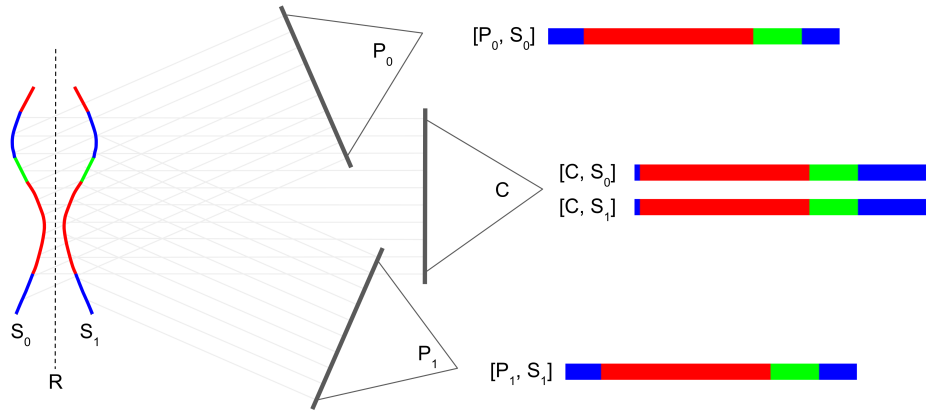


Figure 6.13: Mirror symmetry ambiguity. Under affine projection, a 3D scene (S_0) and its reflection (S_1) across a plane (R) will produce the same image viewed from affine camera C. The consequence of this is that two distinct 3D scenes and camera poses will produce similar images. In this illustration, scene S_0 viewed from camera P_0 will produce the same image as the reflected scene S_1 viewed from P_1. While this relationship is exact in the affine model, we observe that the mini-scene configuration with respect to the scene structure is often well-approximated as affine and training can converge to the near-symmetric solutions. Our LU-NeRF model is explicitly designed to anticipate this failure mode. This illustration is inspired by a similar diagram in [165].

Global pose and scene refinement. After pose averaging, the global pose estimates are expected to be good enough such that any method that requires cameras initialized close to the ground truth should work (*e.g.*, BARF [126], GARF [40]). We apply BARF [126] at this step, which results in both accurate poses and a scene representation accurate enough for realistic novel view synthesis. We refer to the full pipeline as LU-NeRF+Sync.

6.2.4 Experiments

Our method as described in Sec. 6.2.3 starts from a set of mini-scenes that covers the input scene. We evaluate different approaches to constructing mini-scenes, each with different assumptions on the input.

The most strict assumption is that we have an *optimal graph* connecting each image to its nearest neighbors in camera pose space. While this seems unfeasible

	Chair		Hotdog		Lego		Mic		Drums		Ship	
	rot	trans	rot	trans	rot	trans	rot	trans	rot	trans	rot	trans
COLMAP	0.12	0.01	1.24	0.04	2.29	0.10	8.37	0.18	5.91	0.28	0.17	0.01
+BARF	0.14	0.01	1.20	0.01	1.88	0.09	3.73	0.15	8.71	0.54	0.15	0.01
VMRF 120°	4.85	0.28	–	–	2.16	0.16	1.39	0.07	1.28	0.08	16.89	0.71
GNeRF 90°	0.36	0.02	2.35	0.12	0.43	0.02	1.87	0.03	0.20	0.01	3.72	0.18
GNeRF 120°	4.60	0.16	17.19	0.74	4.00	0.20	2.44	0.08	2.51	0.11	31.56	1.38
GNeRF 150°	16.10	0.76	23.53	0.92	4.17	0.36	3.65	0.26	5.01	0.18	–	–
GNeRF 180° (2DOF)	24.46	1.22	36.74	1.46	8.77	0.53	12.96	0.66	**9.01**	0.49	–	–
Ours (3DOF)	**2.64**	**0.09**	**0.24**	**0.01**	**0.09**	**0.00**	**6.68**	**0.10**	12.39	**0.23**	–	–

Table 6.5: Camera pose estimation on unordered image collection. GNeRF [145] and VMRF [278] constrain the elevation range, where the maximum elevation is always 90°. For example, GNeRF 120° only samples elevations in [−30°, 90°]. The 180° variations don't constrain elevation and are closest to our method, but they are still limited to 2 degrees of freedom for assuming upright cameras. Bold numbers indicate superior performance between the bottom two rows, which are the fairest comparison among NeRF-based methods, although our method is still solving a harder 3DOF problem versus 2DOF of GNeRF. We outperform GNeRF in all but one scene in this comparison. COLMAP [196] results in its best possible scenario are shown for reference (higher resolution images and assuming optimal graph to set unregistered poses to the closest registered pose). COLMAP+BARF runs a BARF refinement on top of these initial results, and even in this best-case scenario, our method still outperforms it in some scenes, which shows that LU-NeRF can complement COLMAP and work in scenes COLMAP fails. Our model fails on the Ship scene due to outliers in the connected graph; GNeRF with fewer constraints also fails on it.

in practice, some real-life settings approximate this, for example, when images are deliberately captured in a pattern such as a grid, or if they are captured with camera arrays.

In a less constrained version of the problem, we assume an *ordered image collection*, where the images form a sequence, from where a line graph is trivially built. This is a mild assumption that is satisfied by video data, as well as the common setting of a camera physically moving around a scene sequentially capturing images.

In the most challenging setting, we assume nothing about the scene and only take an *unordered image collection*.

Building graphs from unordered image collections. We evaluate two simple ways of building graphs from unordered image collections. The first is to use deep features from a self-supervised model trained on large image collections. We use the

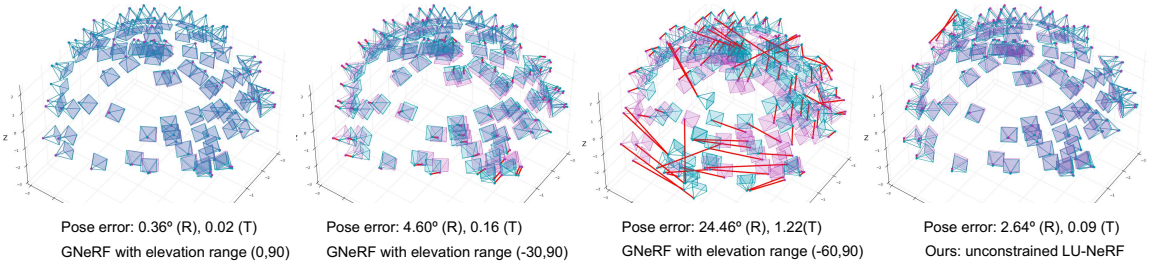


Figure 6.14: Camera pose estimation on unordered image collections. The performance of GNeRF drops dramatically when the pose prior is expanded beyond the true distribution. In comparison, our method does not rely on any prior knowledge of pose distribution.

	Chair			Drums			Lego			Mic		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
GNeRF 90°	31.30	0.95	0.08	24.30	0.90	0.13	28.52	0.91	0.09	31.07	0.96	0.06
GNeRF 120°	25.01	0.89	0.15	20.63	0.86	0.20	22.95	0.85	0.16	23.68	0.93	0.11
GNeRF 150°	22.18	0.88	0.20	19.05	0.83	0.27	21.39	0.84	0.18	23.22	0.92	0.13
VMRF 120°	26.05	0.90	0.14	23.07	0.89	0.16	25.23	0.89	0.12	27.63	0.95	0.08
VMRF 150°	24.53	0.90	0.17	21.25	0.87	0.21	23.51	0.86	0.14	24.39	0.94	0.10
GNeRF 180° (2DOF)	21.27	0.87	0.23	18.08	0.81	0.33	18.22	0.82	0.24	17.22	0.86	0.32
VMRF 180° (2DOF)	23.18	0.89	0.16	20.01	0.84	0.29	21.59	0.83	0.18	20.29	0.90	0.22
Ours (3DOF)	**30.57**	**0.95**	**0.05**	**23.53**	**0.89**	**0.12**	**28.29**	**0.92**	**0.06**	**22.58**	**0.91**	**0.08**

Table 6.6: Novel view synthesis on unordered collections. Our method outperforms the baselines on most scenes while being more general for considering arbitrary rotations with 3 degrees-of-freedom. Here we quote the baseline results from VMRF [278], where *hotdog* is not available.

off-the-shelf DINO model [4, 27] to extract image features and build the graph based on the cosine distance between these features. The second is to simply use the ℓ_1 distance in pixel space against slightly shifted and rotated versions of the images. Neither of these approaches is ideal. The deep features are typically coarse and too general, failing to detect specific subtle changes on the scene. The ℓ_1 distance has the opposite issue, where small changes can result in large distances. Exploring other methods for finding a proxy metric for the relative pose in image space is a direction for future work.

Datasets. We compare with prior works on the synthetic-NeRF dataset [149]. We use the training split of the original dataset as our *unordered image collection* which consists of 100 unordered images per 3D scene. We use the first 8 images from

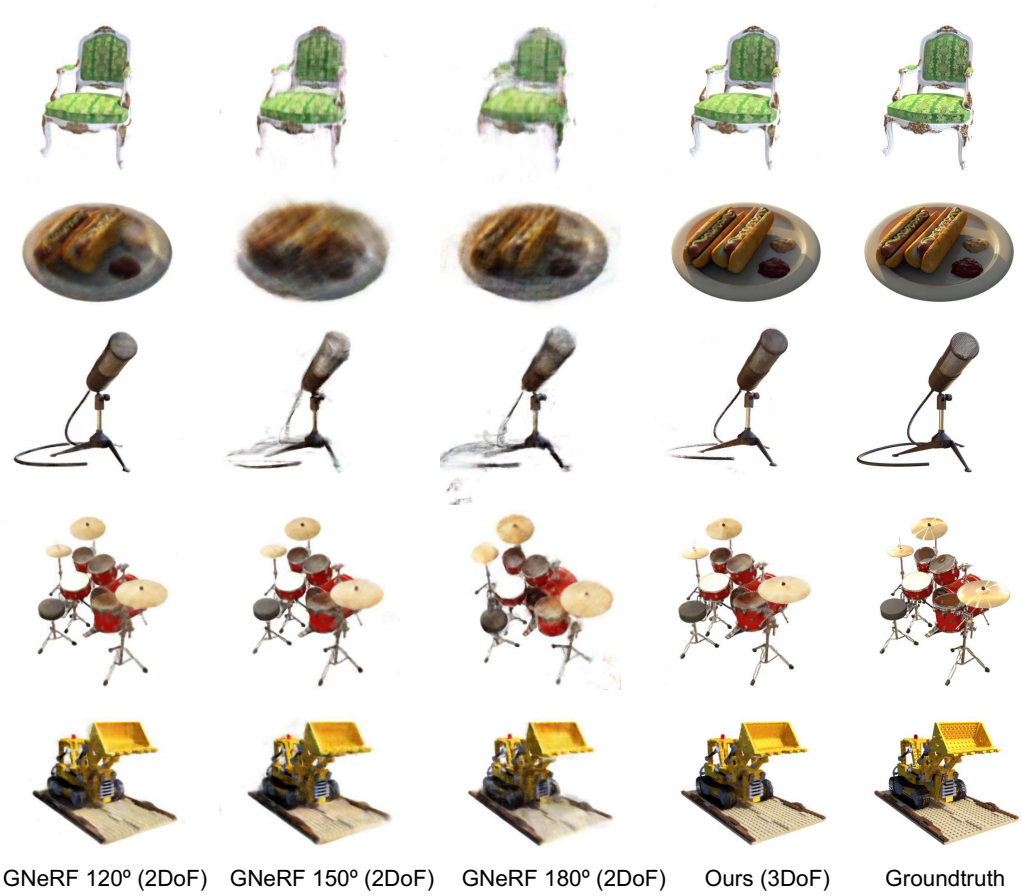


Figure 6.15: Novel view synthesis on unordered image collections. GNeRF makes assumptions on the elevation range, where the maximum elevation is always 90°. For instance, GNeRF 150° only samples elevations in [-60°, 90°]. The 180° variations don't constrain elevation and are closest to our method, but they are still limited to 2 degrees of freedom for assuming upright cameras. The performance of GNeRF drops as prior poses are less constrained. Please zoom into the figure to see the details in the renderings.

the validation set as our test set for the novel view synthesis task, following prior works [145, 278]

To evaluate on image sequences, where the order of images is known, we further render a Blender *ordered image collection* with 100 images along a spiral path per scene. The images are resized to 400×400 in our experiments.

We also evaluate on real images from the object-centric videos in Objectron [3]. The dataset provides ground truth poses computed using AR solutions at 30fps, and

Image size	Chair	Hotdog	Lego	Mic	Drums	Ship
400×400	100	88	100	15	74	45
800×800	100	98	100	80	84	100

Table 6.7: Number of images registered by COLMAP on Blender.

we construct a wider-baseline dataset by subsampling every 15th frame and selecting videos with limited texture (Fig. 6.17).

Evaluation metrics. We evaluate the tasks of camera pose estimation and novel view synthesis. For camera pose estimation, we report the camera rotation and translation error using Procrustes analysis as in BARF [126]. For novel view synthesis, we report the PSNR, SSIM, and LPIPS [281].

Baseline methods. We compare with GNeRF [145], VMRF [278], and COLMAP [196] throughout our experiments. GNeRF samples camera poses from a predefined prior pose distribution and trains a GAN-based neural rendering model to build the correspondence between the sampled camera poses and 2D renderings. The method provides accurate pose estimation under *proper* prior pose distribution. However, its performance degrades significantly when the prior pose distribution doesn't match the groundtruth. VMRF attempts to relieve the reliance of GNeRF on the prior pose distribution but still inherits its limitations. In our experiments, we evaluate with the default pose priors of GNeRF on the NeRF-synthetic dataset, azimuth $\in [0°, 360°]$ and elevation $\in [0°, 90°]$, and also on less constrained cases. COLMAP works reliably in texture-rich scenes but may fail dramatically on texture-less surfaces.

Implementation details. We use a compact network for LU-NeRF to speed up the training and minimize the memory cost. Specifically, we use a 4-layer MLP without positional encoding and conditioning on the view directions. We stop the training early when the change of camera poses on mini-scenes is under a predefined threshold. To resolve the mirror symmetry ambiguity (Sec. 6.2.3.2), we train two additional LU-

	Chair		Drums		Lego		Materials		Mean	
	rot	trans	rot	trans	rot	trans	rot	trans	rot	trans
GNeRF 90°	11.6	0.49	8.03	0.29	7.89	0.19	6.80	0.12	8.91	0.30
GNeRF 180°	27.7	1.17	130	6.23	123	4.31	30.9	1.40	94.9	3.27
Ours (3DOF)	**0.72**	**0.03**	**0.07**	**0.08**	**1.96**	**0.00**	**0.31**	**0.00**	**0.76**	**0.03**

Table 6.8: Pose estimation on the Blender *ordered image collections*. We report rotation errors in degrees and translation at the input scene scale. Our method can be more easily applied to ordered image collections since the graph-building step becomes trivial. In this case, we outperform GNeRF even when it is aided by known and constrained pose distributions.

NeRFs for a fixed number of training iterations (50k by default). The weight of the depth regularization is 10 times larger than the photometric ℓ_2 loss throughout our experiments.

6.2.4.1 Unordered Image Collections

	Chair			Drums			Lego			Materials		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
GNeRF 90°	27.22	0.93	0.17	20.88	0.84	0.29	22.83	0.83	0.25	22.58	0.85	0.20
GNeRF 180° (2DOF)	23.50	0.91	0.26	11.01	0.81	0.56	9.78	**0.78**	0.53	9.48	0.65	0.50
Ours (3DOF)	**33.94**	**0.98**	**0.03**	**25.29**	**0.91**	**0.08**	**15.90**	0.72	**0.20**	**29.73**	**0.96**	**0.03**

Table 6.9: Novel view synthesis on Blender *ordered image collections*. The relative improvement of our method with respect to GNeRF is larger with an ordered image collection, since we avoid the difficult step of building the initial graph.

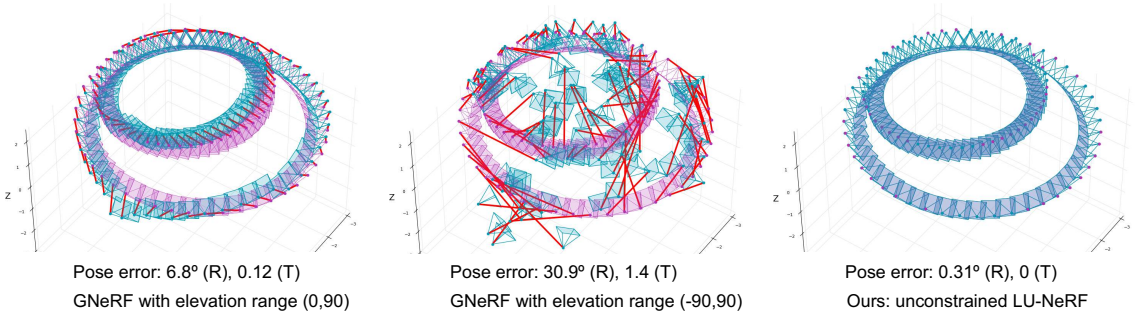


Figure 6.16: Pose estimation on the Blender Materials *ordered image collection*. The performance of GNeRF degrades with unconstrained elevation (left vs. middle). The proposed method achieves accurate pose estimation without assumptions on the prior pose distribution.

Camera pose estimation. Tab. 6.5 compares our method to GNeRF, VMRF, and COLMAP in the camera pose estimation task. GNeRF achieves high pose estimation accuracy when the elevation angles are uniformly sampled from a 90° interval; however, its performance drops significantly when the range of elevation is enlarged. Our method outperforms GNeRF in most scenes when the prior pose distribution is unknown, since we do not require any prior knowledge of the camera poses. Fig. 6.14 provides the visualization of the estimated camera poses from GNeRF under different prior pose distributions and our method.

Tab. 6.7 shows the number of images COLMAP registers out of 100 in each scene. COLMAP is sensitive to image resolution, and its performance drops significantly on low-resolution images. For instance, COLMAP only registers 15 images out of 100 on the Mic scene when the image size is 400×400. Our method provides accurate pose estimation for all cameras given 400×400 images. Tab. 6.5 also reports how COLMAP performs in the pose estimation task on the Blender scenes. We use the most favorable settings for COLMAP – 800×800 images and set the poses of unregistered cameras to the poses of the nearest registered camera, assuming the *optimal graph* is known, while our method makes no such assumption. Nevertheless, our model achieves better performance than COLMAP in some scenes, even when a BARF refinement is applied to initial COLMAP results. This shows that LU-NeRF complements COLMAP by working in scenes where COLMAP fails.

Novel view synthesis. Fig. 6.15 and Tab. 6.6 show our results in the task of novel view synthesis on unordered image collections. The results are consistent with the quantitative pose evaluation – our model outperforms both VMRF and GNeRF when no priors on pose distribution are assumed.

6.2.4.2 Ordered Image Collections

Blender. Tab. 6.8, Tab. 6.9, and Fig. 6.16 summarize the results on the Blender *ordered image collection*. Our method outperforms GNeRF with both constrained and unconstrained pose distributions even though the elevation of the cameras in this dataset is constrained. Our method utilizes the image order to build a connected graph and does not make any assumptions about the camera distribution. Results in Tab. 6.9 show that the view synthesis results are in sync with the pose estimation results. GNeRF degrades significantly under unconstrained pose priors, while our method outperforms GNeRF consistently across different scenes.

Objectron. We further compare with COLMAP on real images from the Objectron dataset. COLMAP can be improved with modern feature extraction and matching algorithms [190] such as SuperPoint [56] and SuperGLUE [191] (denoted COLMAP-SPSG), or LoFTR [213] (denoted COLMAP-LoFTR), but these still struggle in scenes with little or repeated texture. Tab. 6.10 and Fig. 6.17 show our results *without BARF refinement* on difficult scenes from Objectron.

Figure 6.17: Camera pose estimation on textureless scenes. COLMAP fails to register any cameras in these Objectron scenes. Ground truth cameras are in purple, our predictions in blue.

6.2.4.3 Analysis

This section provides additional analysis of our approach. All the experiments discussed below were conducted on the unordered image collection.

Mirror symmetry ambiguity. Tab. 6.11 shows the performance of our full method with and without the proposed solution to the mirror-symmetry ambiguity (Sec. 6.2.3.2).

	Bike	Chair	Cup	Laptop	Shoe	Book
Rotation:						
COLMAP	–	17.2	–	–	14.1	–
COLMAP-SPSG	129	28.3	–	–	**8.3**	–
COLMAP-LoFTR	**1.1**	6.7	6.3	**9.5**	14.5	83.4
Ours	15.6	**2.6**	**6.1**	17.8	8.8	**3.2**
Translation:						
COLMAP	–	0.04	–	–	**0.03**	–
COLMAP-SPSG	1.71	0.12	–	–	0.04	–
COLMAP-LoFTR	**0.10**	0.07	**0.03**	0.34	0.14	0.67
Ours	0.13	**0.03**	0.11	**0.16**	0.20	**0.03**

Table 6.10: Comparison with COLMAP on Objectron [3]. We report rotation (°) and translation errors on select scenes from Objectron that are challenging to COLMAP. "–" denotes failure to estimate any camera poses. **COLMAP-SPSG** is an improved version [190] with SuperPoint [56] and SuperGLUE [191] as descriptor and matcher, respectively. **COLMAP-LoFTR** improves COLMAP with LoFTR [213], a detector-free feature matcher. Translation errors are in the scale of the ground truth scene.

Resolving the ambiguity improves performance consistently, confirming the importance of this component to our pipeline. For closer inspection, we present qualitative results for LU-NeRF with and without ambiguity resolution for select mini-scenes in Fig. 6.18. Fig. 6.18 presents a visual comparison between LU-NeRF with and without the proposed solution to the mirror-symmetry ambiguity. Without the ambiguity resolution, the predicted depths are reflected across a plane parallel to the image plane (having the effect of inverted disparity maps), and the poses are reflected across the center camera of a mini-scene. Our LU-NeRF-2 rectifies the predicted geometry and local camera poses, which effectively resolves the ambiguity.

6.2.5 Conclusion

In this work, we propose to estimate the neural scene representation and camera poses jointly from an unposed image collection through a process of synchronizing local unposed NeRFs. Unlike prior works, our method does not rely on a proper

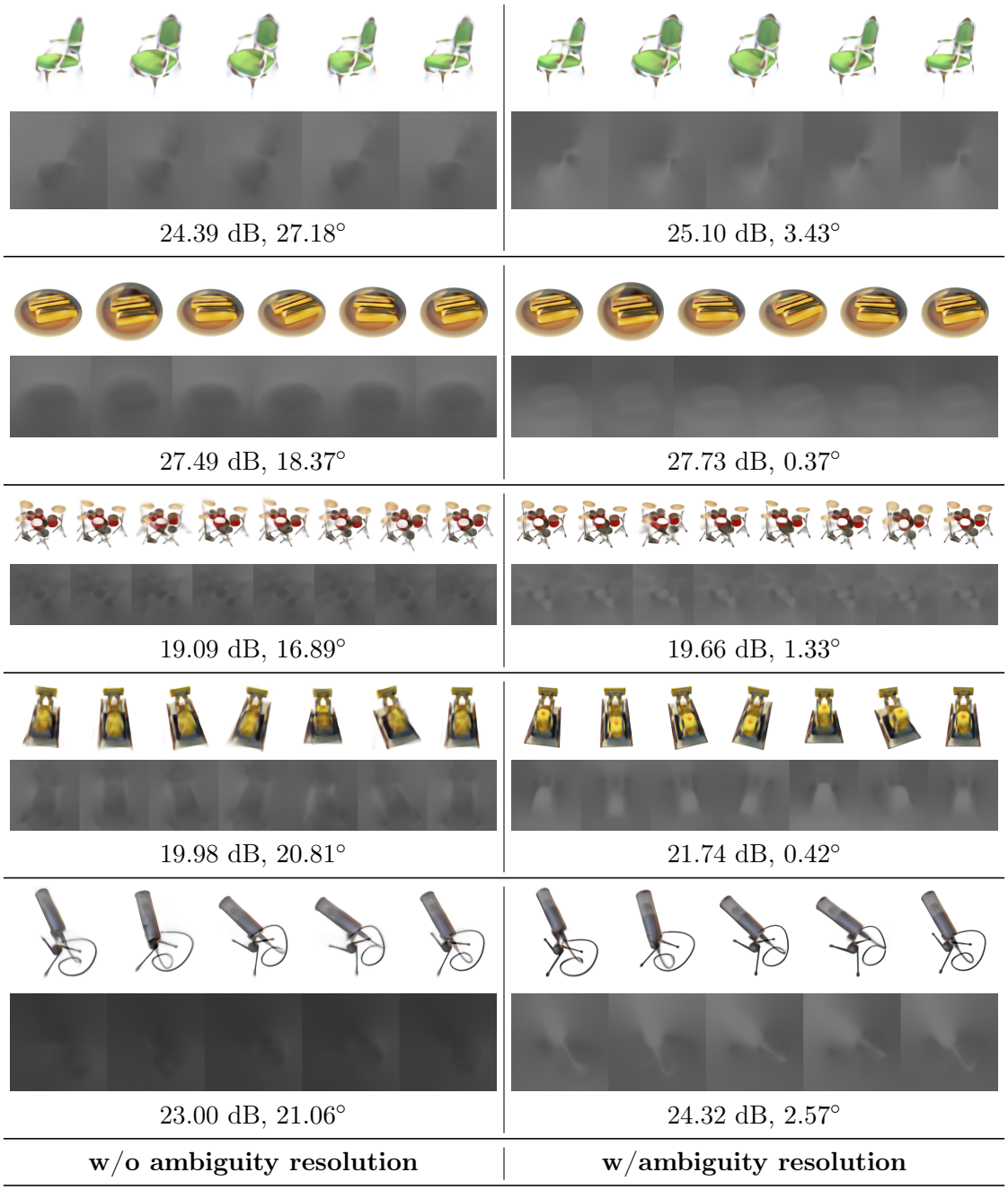


Figure 6.18: Mirror symmetry ambiguity. For specific mini-scenes, we present renderings, disparity maps, PSNRs between the renderings and the groundtruth, and relative rotation errors (*lower is better*) for LU-NeRF with and without the proposed solution to the mirror-symmetry ambiguity. Brightness is inversely related to depth in the disparity map. The groundtruth depth maps are not available with the dataset.

prior pose distribution and is flexible enough to operate in general **SE**(3) pose settings. Our framework works reliably in low-texture or low-resolution images and thus

Ambiguity	Chair	Hotdog	Lego	Mic	Drums
w/o resolution	39.14	138.9	0.48	107.9	11.35
w/ resolution	**4.24**	**0.23**	**0.07**	**0.84**	**0.05**

Table 6.11: Mirror symmetry ambiguity. The mean rotation error in degrees for our pipeline (starting with the optimal graph), with and without the proposed strategy to resolve the ambiguity.

complements the feature-based SfM algorithms. Our pipeline also naturally exploits sequential image data, which is easy to acquire in practice.

One limitation of our method is the computational cost, which can be relieved by recent advances in neural rendering [221]. Another limitation is the difficulty in building graphs for unordered scenes, which is a promising direction for future work.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

This dissertation addresses a fundamental challenge of deep learning — the dependency on costly human supervision in terms of data collection, annotations, and neural network architecture design. We have developed a variety of label-efficient algorithms geared towards landmark detection and pose estimation. Furthermore, we have proposed a learning approach that is robust to noise for object detection and demonstrated its applicability in large-scale ecological studies. From a theoretical standpoint, we have explored the understanding of deep neural network architectures, viewing them through the lens of Gaussian processes. Lastly, our efforts in the realm of 3D generation and manipulation have also been thoroughly reviewed within the dissertation.

7.2 Future work

Towards the goal of building intelligent agents capable of understanding and interacting with the 3D visual world, I aim to explore the following directions in the near future.

Holistic 3D scene understanding and reconstruction. We humans have a holistic understanding of the 3D visual world — we can easily perceive the object categories, their location, and shapes and even interact with them. This is a fundamental capability required of intelligent agents to navigate and interact with the 3D environment. Besides this, reconstructing a realistic and immersive virtual 3D world

has many applications in VR/AR, robotics, and autonomous driving. However, such holistic 3D scene understanding and generation are beyond the current state-of-the-art computer vision systems. There are several critical challenges to address. First, scene understanding and 3D reconstruction are usually studied separately, which I believe should be integrated in a way that they are mutually beneficial; Second, compared to 2D tasks, the lack of human annotations becomes even more problematic for 3D tasks (*e.g.*, 3D object detection and segmentation, 3D pose estimation); Third, unlike 2D images, 3D models are expensive to acquire, especially for deformable objects such as animals; I aim to build a system to holistically understand and reconstruct the 3D scene with minimal human supervision. I plan to design end-to-end models that learn visual understanding and 3D reconstruction modules simultaneously; I will leverage video datasets to learn the shape and structure of object categories and advances in neural rendering to perform joint 2D/3D reasoning. I will also systematically evaluate the state-of-the-art self-supervised learning (SSL) methods (*e.g.*, masked auto-encoding) for 3D vision tasks and explore more effective SSL methods.

Multi-modal machine perception. Humans learn to perceive the physical world through multi-sensory systems — vision, audition, touch, smell, *etc.* These modalities are overlapped and temporally aligned and thus can supervise each other. I envision a future where intelligent agents equipped with multiple sensors could learn to understand the world by simply perceiving and acting in the world. Multi-modal perception not only plays a fundamental role in the development of human intelligence but also becomes increasingly important in real applications. For example, data collection capability in ecological research has been drastically increased by the recent development of sensory techniques, such as remote sensing, camera traps, and acoustic sensors. However, there is a mismatch between the ever-growing multi-sensory archives and our ability to distill biological information from the data collections. I plan to build generic network architectures that handle data from different sensors

and can be trained with correspondence among different modalities. I will also collaborate with experts from other disciplines, such as NLP, robotics, and ecology, to develop machine learning tools to analyze domain-specific multi-sensory data.

Learning to see in the wild. The field of computer vision has been driven by machine learning models trained on massive data collections. Despite the significant progress in the past decade, the literature mainly focuses on training and evaluating models on *curated benchmarks* (*e.g.*, ImageNet). However, the curated datasets are only a limited fraction of the general data distribution. This raises the concern that our progress in controlled settings may not be applicable in real-life applications (*e.g.*, autonomous driving and mobile home assistants). Manually curating datasets becomes even more prohibitive for self-supervised and semi-supervised learning on the sheer volume of unlabeled images or videos. Our prior work [211] shows that existing semi-supervised learning methods do not work out-of-the-box in realistic benchmarks where data exhibits a long-tailed distribution of fine-grained categories. Inspired by this work, I will strive to train and evaluate machine learning models *in the wild*. I plan to build new benchmarks that match the realistic data distribution and then systematically assess prior works on the new datasets. I will also design novel algorithms to tackle the challenges conveyed by these realistic evaluations.

Multidisciplinary collaborations. I've established broad collaborations with researchers from the industry (*e.g.*, Google, Adobe, and Snap) to tackle both fundamental and applied research problems related to Graphics [222] and Robotics [223**?**]. I've also been collaborating with ecologists from Cornell Lab of Ornithology and Colorado State University to solve challenges in ecology with AI techniques [17, 55, 174, 224, 227], as well as chemists from the Chemical Engineering Department at UMass Amherst to discover novel materials [135]. I'll continue such multidisciplinary collaborations and promote the application of AI in different scientific fields.

BIBLIOGRAPHY

[1] Abdal, Rameen, Qin, Yipeng, and Wonka, Peter. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV* (2019), pp. 4432–4441.

[2] Achille, Alessandro, and Soatto, Stefano. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research* (2018).

[3] Ahmadyan, Adel, Zhang, Liangkai, Ablavatski, Artsiom, Wei, Jianing, and Grundmann, Matthias. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR* (2021), pp. 7822–7831.

[4] Amir, Shir, Gandelsman, Yossi, Bagon, Shai, and Dekel, Tali. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814 2*, 3 (2021), 4.

[5] An, Xiaobo, Tong, Xin, Denning, Jonathan D, and Pellacini, Fabio. Appwarp: Retargeting measured materials by appearance-space warping. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (2011), pp. 1–10.

[6] Arandjelovic, Relja, Gronat, Petr, Torii, Akihiko, Pajdla, Tomas, and Sivic, Josef. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR* (2016).

[7] Arrigoni, Federica, Rossi, Beatrice, and Fusiello, Andrea. Spectral synchronization of multiple views in se(3). *SIAM Journal on Imaging Sciences* (2016).

[8] Athar, ShahRukh, Burnaev, Evgeny, and Lempitsky, Victor. Latent convolutional models. In *ICLR* (2018).

[9] Aygun, Mehmet, and Mac Aodha, Oisin. Demystifying unsupervised semantic correspondence estimation. In *ECCV* (2022).

[10] Bachman, Philip, Hjelm, R Devon, and Buchwalter, William. Learning representations by maximizing mutual information across views. In *NeurIPS* (2019).

[11] Baranchuk, Dmitry, Voynov, Andrey, Rubachev, Ivan, Khrulkov, Valentin, and Babenko, Artem. Label-efficient semantic segmentation with diffusion models. In *ICLR* (2022).

[12] Barron, Jonathan T, Mildenhall, Ben, Tancik, Matthew, Hedman, Peter, Martin-Brualla, Ricardo, and Srinivasan, Pratul P. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV* (2021).

[13] Barron, Jonathan T., Mildenhall, Ben, Verbin, Dor, Srinivasan, Pratul P., and Hedman, Peter. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR* (2022).

[14] Bau, David, Liu, Steven, Wang, Tongzhou, Zhu, Jun-Yan, and Torralba, Antonio. Rewriting a deep generative model. In *ECCV* (2020), Springer, pp. 351–369.

[15] Bau, David, Strobelt, Hendrik, Peebles, William, Zhou, Bolei, Zhu, Jun-Yan, Torralba, Antonio, et al. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727* (2020).

[16] Belhumeur, Peter N, Kriegman, David J, and Yuille, Alan L. The bas-relief ambiguity. *IJCV*.

[17] Belotti, Maria Carolina TD, Deng, Yuting, Zhao, Wenlong, Simons, Victoria F, Cheng, Zezhou, Perez, Gustavo, Tielens, Elske, Maji, Subhransu, Sheldon, Daniel R, Kelly, Jeffrey F, et al. Long-term analysis of persistence and size of swallow and martin roosts in the us great lakes. *Remote Sensing in Ecology and Conservation* (2022).

[18] Bespalov, Iaroslav, Buzun, Nazar, and Dylov, Dmitry V. Brulé: Barycenter-regularized unsupervised landmark extraction. *arXiv preprint arXiv:2006.11643* (2020).

[19] Bian, Wenjing, Wang, Zirui, Li, Kejie, Bian, Jia-Wang, and Prisacariu, Victor Adrian. Nope-nerf: Optimising neural radiance field with no pose prior. *arXiv preprint arXiv:2212.07388* (2022).

[20] Blei, David M, Kucukelbir, Alp, and McAuliffe, Jon D. Variational inference: A review for statisticians. *Journal of the American Statistical Association 112*, 518 (2017).

[21] Borovykh, Anastasia. A Gaussian Process Perspective on Convolutional Neural Networks. *arXiv:1810.10798* (2018).

[22] Boss, Mark, Engelhardt, Andreas, Kar, Abhishek, Li, Yuanzhen, Sun, Deqing, Barron, Jonathan T., Lensch, Hendrik P.A., and Jampani, Varun. SAMURAI: Shape And Material from Unconstrained Real-world Arbitrary Image collections. In *NeurIPS* (2022).

[23] Bridge, Eli S, Pletschet, Sandra M, Fagin, Todd, Chilson, Phillip B, Horton, Kyle G, Broadfoot, Kyle R, and Kelly, Jeffrey F. Persistence and habitat associations of purple martin roosts quantified via weather surveillance radar. *Landscape ecology 31*, 1 (2016).

[24] Brodley, Carla E, and Friedl, Mark A. Identifying mislabeled training data. *Journal of artificial intelligence research* (1999).

[25] Burt, Peter J, and Adelson, Edward H. The laplacian pyramid as a compact image code. In *Readings in computer vision*. Elsevier, 1987, pp. 671–679.

[26] Caron, Mathilde, Touvron, Hugo, Misra, Ishan, Jégou, Hervé, Mairal, Julien, Bojanowski, Piotr, and Joulin, Armand. Emerging properties in self-supervised vision transformers. In *ICCV* (October 2021), pp. 9650–9660.

[27] Caron, Mathilde, Touvron, Hugo, Misra, Ishan, Jégou, Hervé, Mairal, Julien, Bojanowski, Piotr, and Joulin, Armand. Emerging properties in self-supervised vision transformers. In *ICCV* (2021).

[28] Chang, Angel X, Funkhouser, Thomas, Guibas, Leonidas, Hanrahan, Pat, Huang, Qixing, Li, Zimo, Savarese, Silvio, Savva, Manolis, Song, Shuran, Su, Hao, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

[29] Chatfield, Ken, Simonyan, Karen, Vedaldi, Andrea, and Zisserman, Andrew. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531* (2014).

[30] Chen, Anpei, Xu, Zexiang, Zhao, Fuqiang, Zhang, Xiaoshuai, Xiang, Fanbo, Yu, Jingyi, and Su, Hao. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *CVPR* (2021).

[31] Chen, Di, Liu, Yu, Huang, Lianghua, Wang, Bin, and Pan, Pan. GeoAug: Data augmentation for few-shot nerf with geometry constraints. In *ECCV* (2022).

[32] Chen, Kevin, Choy, Christopher B, Savva, Manolis, Chang, Angel X, Funkhouser, Thomas, and Savarese, Silvio. Text2shape: Generating shapes from natural language by learning joint embeddings. In *ACCV* (2018), Springer, pp. 100–116.

[33] Chen, Shu, Zhang, Yang, Xu, Yaxin, and Zou, Beiji. Structure-aware nerf without posed camera via epipolar constraint. *CoRR abs/2210.00183* (2022).

[34] Chen, Ting, Kornblith, Simon, Norouzi, Mohammad, and Hinton, Geoffrey. A simple framework for contrastive learning of visual representations. *ICML* (2020).

[35] Chen, Ting, Kornblith, Simon, Swersky, Kevin, Norouzi, Mohammad, and Hinton, Geoffrey. Big self-supervised models are strong semi-supervised learners. *NeurIPS* (2020).

[36] Chen, Xinlei, Fan, Haoqi, Girshick, Ross, and He, Kaiming. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[37] Chen, Zhiqin, and Zhang, Hao. Learning implicit fields for generative shape modeling. In *CVPR* (2019), pp. 5939–5948.

[38] Cheng, Zezhou, Su, Jong-Chyi, and Maji, Subhransu. On equivariant and invariant learning of object landmark representations. In *ICCV* (2021), pp. 9897–9906.

[39] Chilson, Carmen, Avery, Katherine, McGovern, Amy, Bridge, Eli, Sheldon, Daniel, and Kelly, Jeffrey. Automated detection of bird roosts using nexrad radar data and convolutional neural networks. *Remote Sensing in Ecology and Conservation 5*, 1 (2019).

[40] Chng, Shin-Fang, Ramasinghe, Sameera, Sherrah, Jamie, and Lucey, Simon. GARF: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. In *ICCV* (2021).

[41] Cho, Youngmin, and Saul, Lawrence K. Kernel Methods for Deep Learning. In *NeurIPS* (2009), pp. 342–350.

[42] Choi, Yunjey, Choi, Minje, Kim, Munyoung, Ha, Jung-Woo, Kim, Sunghun, and Choo, Jaegul. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR* (2018), pp. 8789–8797.

[43] Choy, Christopher B, Xu, Danfei, Gwak, JunYoung, Chen, Kevin, and Savarese, Silvio. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV* (2016), Springer, pp. 628–644.

[44] Chum, Ondrej, Werner, Tomás, and Matas, Jiri. Two-view geometry estimation unaffected by a dominant plane. In *CVPR* (2005).

[45] Cimpoi, Mircea, Maji, Subhransu, and Vedaldi, Andrea. Deep filter banks for texture recognition and segmentation. In *CVPR* (2015).

[46] Cohen, Taco, and Welling, Max. Group equivariant convolutional networks. In *ICML* (2016).

[47] Collins, Edo, Achanta, Radhakrishna, and Susstrunk, Sabine. Deep feature factorization for concept discovery. In *ECCV* (2018).

[48] Crum, Timothy D, and Alberty, Ron L. The WSR-88D and the WSR-88D operational support facility. *Bulletin of the American Meteorological Society 74*, 9 (1993).

[49] Dabov, Kostadin, Foi, Alessandro, Katkovnik, Vladimir, and Egiazarian, Karen. Image Denoising by Sparse 3-D Transform-domain Collaborative Filtering. *IEEE Transactions on image processing 16*, 8 (2007), 2080–2095.

[50] DeCarlo, Doug, Finkelstein, Adam, Rusinkiewicz, Szymon, and Santella, Anthony. Suggestive contours for conveying shape. In *ACM SIGGRAPH 2003 Papers*. 2003, pp. 848–855.

[51] Delanoy, Johanna, Aubry, Mathieu, Isola, Phillip, Efros, Alexei A, and Bousseau, Adrien. 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1*, 1 (2018), 1–22.

[52] Dellaert, Frank, Rosen, David M., Wu, Jing, Mahony, Robert, and Carlone, Luca. Shonan rotation averaging: Global optimality by surfing $so(p)^n$. In *ECCV* (2020).

[53] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *CVPR* (2009).

[54] Deng, Kangle, Liu, Andrew, Zhu, Jun-Yan, and Ramanan, Deva. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR* (June 2022).

[55] Deng, Yuting, Belotti, Maria Carolina TD, Zhao, Wenlong, Cheng, Zezhou, Perez, Gustavo, Tielens, Elske, Simons, Victoria F, Sheldon, Daniel R, Maji, Subhransu, Kelly, Jeffrey F, et al. Quantifying long-term phenological patterns of aerial insectivores roosting in the great lakes region using weather surveillance radar. *Global Change Biology* (2022).

[56] DeTone, Daniel, Malisiewicz, Tomasz, and Rabinovich, Andrew. Superpoint: Self-supervised interest point detection and description. In *CVPRW* (2018), pp. 224–236.

[57] Divon, Gilad, and Tal, Ayellet. Viewpoint estimation—insights & model. In *ECCV* (2018), pp. 252–268.

[58] Dokter, Adriaan M., Desmet, Peter, Spaaks, Jurriaan H., van Hoey, Stijn, Veen, Lourens, Verlinden, Liesbeth, Nilsson, Cecilia, Haase, Günther, Leijnse, Hidde, Farnsworth, Andrew, Bouten, Willem, and Shamoun-Baranes, Judy. bioRad: biological analysis and visualization of weather radar data. *Ecography* (2018).

[59] Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[60] Dosovitskiy, Alexey, Springenberg, Jost Tobias, Riedmiller, Martin, and Brox, Thomas. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS* (2014).

[61] Esteves, Carlos, Allen-Blanchette, Christine, Makadia, Ameesh, and Daniilidis, Kostas. Learning so(3) equivariant representations with spherical cnns. In *ECCV* (2018), pp. 52–68.

[62] Esteves, Carlos, Makadia, Ameesh, and Daniilidis, Kostas. Spin-weighted spherical cnns. *NeurIPS 33* (2020), 8614–8625.

[63] Everingham, Mark, Van Gool, Luc, Williams, Christopher KI, Winn, John, and Zisserman, Andrew. The pascal visual object classes (voc) challenge. *IJCV 88*, 2 (2010), 303–338.

[64] Fan, Haoqiang, Su, Hao, and Guibas, Leonidas J. A point set generation network for 3d object reconstruction from a single image. In *CVPR* (2017), pp. 605–613.

[65] Fang, Hao-Shu, Wang, Chenxi, Gou, Minghao, and Lu, Cewu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *CVPR* (2020), pp. 11444–11453.

[66] Feng, Zhen-Hua, Kittler, Josef, Awais, Muhammad, Huber, Patrik, and Wu, Xiao-Jun. Wing loss for robust facial landmark localisation with convolutional neural networks. In *CVPR* (2018).

[67] Fu, Jianlong, Zheng, Heliang, and Mei, Tao. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR* (2017).

[68] Gadelha, Matheus, Wang, Rui, and Maji, Subhransu. Deep manifold prior. In *ICCV* (2021), pp. 1107–1116.

[69] Garriga-Alonso, Adrià, Aitchison, Laurence, and Rasmussen, Carl Edward. Deep Convolutional Networks as Shallow Gaussian Processes. *arXiv:1808.05587* (2018).

[70] Geiger, Andreas, Lenz, Philip, and Urtasun, Raquel. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR* (2012), IEEE, pp. 3354–3361.

[71] Gens, Robert, and Domingos, Pedro M. Deep symmetry networks. In *NeurIPS* (2014).

[72] Ghosh, Aritra, Kumar, Himanshu, and Sastry, PS. Robust loss functions under label noise for deep neural networks. In *AAAI* (2017), pp. 1919–1925.

[73] Gidaris, Spyros, Singh, Praveer, and Komodakis, Nikos. Unsupervised representation learning by predicting image rotations. In *ICLR* (2018).

[74] Gkioxari, Georgia, Malik, Jitendra, and Johnson, Justin. Mesh r-cnn. In *ICCV* (2019), pp. 9785–9795.

[75] Goel, Shubham, Kanazawa, Angjoo, and Malik, Jitendra. Shape and viewpoint without keypoints. In *ECCV* (2020), Springer, pp. 88–104.

[76] Gonzalez-Garcia, Abel, Modolo, Davide, and Ferrari, Vittorio. Do semantic parts emerge in convolutional neural networks? *IJCV* (2018).

[77] Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative Adversarial Networks. In *NeurIPS* (2014).

[78] Gower, John C. Generalized procrustes analysis. *Psychometrika 40*, 1 (1975), 33–51.

[79] Goyal, Priya, Mahajan, Dhruv, Gupta, Abhinav, and Misra, Ishan. Scaling and benchmarking self-supervised visual representation learning. In *ICCV* (2019).

[80] Grabner, Alexander, Roth, Peter M, and Lepetit, Vincent. 3d pose estimation and 3d model retrieval for objects in the wild. In *CVPR* (2018), pp. 3022–3031.

[81] Grady, Leo. Random walks for image segmentation. *IEEE TPAMI 28*, 11 (2006), 1768–1783.

[82] Gu, Jinjin, Shen, Yujun, and Zhou, Bolei. Image processing using multi-code gan prior. In *CVPR* (2020), pp. 3012–3021.

[83] Guillard, Benoit, Remelli, Edoardo, Yvernay, Pierre, and Fua, Pascal. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *ICCV* (2021).

[84] Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron. Improved training of wasserstein gans. In *NeurIPS* (2017).

[85] Gutmann, Michael, and Hyvärinen, Aapo. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS* (2010).

[86] Hadsell, Raia, Chopra, Sumit, and LeCun, Yann. Dimensionality reduction by learning an invariant mapping. In *CVPR* (2006).

[87] Hao, Zekun, Averbuch-Elor, Hadar, Snavely, Noah, and Belongie, Serge. Dualsdf: Semantic shape manipulation using a two-level representation. In *CVPR* (2020), pp. 7631–7641.

[88] Hariharan, Bharath, Arbeláez, Pablo, Girshick, Ross, and Malik, Jitendra. Hypercolumns for object segmentation and fine-grained localization. In *CVPR* (2015).

[89] Hartley, Richard, Trumpf, Jochen, Dai, Yuchao, and Li, Hongdong. Rotation Averaging. *IJCV 101*, 2 (2013).

[90] Hartley, Richard, and Zisserman, Andrew. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[91] Hartley, Richard, and Zisserman, Andrew. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[92] He, Kaiming, Chen, Xinlei, Xie, Saining, Li, Yanghao, Dollár, Piotr, and Girshick, Ross. Masked autoencoders are scalable vision learners. In *CVPR* (2022), pp. 16000–16009.

[93] He, Kaiming, Fan, Haoqi, Wu, Yuxin, Xie, Saining, and Girshick, Ross. Momentum contrast for unsupervised visual representation learning. In *CVPR* (2020).

[94] He, Kaiming, Fan, Haoqi, Wu, Yuxin, Xie, Saining, and Girshick, Ross. Momentum contrast for unsupervised visual representation learning. In *CVPR* (2020), pp. 9729–9738.

[95] He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, and Girshick, Ross. Mask r-cnn. In *ICCV* (2017), pp. 2961–2969.

[96] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *CVPR* (2016).

[97] Heide, Felix, Heidrich, Wolfgang, and Wetzstein, Gordon. Fast and Flexible Convolutional Sparse Coding. In *Computer Vision and Pattern Recognition (CVPR)* (2015).

[98] Hénaff, Olivier J, Srinivas, Aravind, De Fauw, Jeffrey, Razavi, Ali, Doersch, Carl, Eslami, SM, and Oord, Aaron van den. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272* (2019).

[99] Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS 30* (2017).

[100] Hjelm, R Devon, Fedorov, Alex, Lavoie-Marchildon, Samuel, Grewal, Karan, Bachman, Phil, Trischler, Adam, and Bengio, Yoshua. Learning deep representations by mutual information estimation and maximization. In *ICLR* (2019).

[101] Ho, Jonathan, Jain, Ajay, and Abbeel, Pieter. Denoising diffusion probabilistic models. *NeurIPS 33* (2020), 6840–6851.

[102] Hodan, Tomáš, Haluza, Pavel, Obdržálek, Štepán, Matas, Jiri, Lourakis, Manolis, and Zabulis, Xenophon. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *WACV* (2017), IEEE, pp. 880–888.

[103] Hoen, B. D., Diffendorfer, J. E., Rand, J. T., Kramer, L. A., Garrity, C. P., and Hunt, H.E. United states wind turbine database. *U.S. Geological Survey, American Wind Energy Association, and Lawrence Berkeley National Laboratory data release: USWTDB V1.3.* (2019). `https://eerscmap.usgs.gov/uswtdb`.

[104] Insafutdinov, Eldar, and Dosovitskiy, Alexey. Unsupervised learning of shape and pose with differentiable point clouds. In *NeurIPS* (2018).

[105] Jakab, Tomas, Gupta, Ankush, Bilen, Hakan, and Vedaldi, Andrea. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS* (2018).

[106] Jakab, Tomas, Gupta, Ankush, Bilen, Hakan, and Vedaldi, Andrea. Self-supervised learning of interpretable keypoints from unlabelled videos. In *CVPR* (2020).

[107] Jiang, Huaizu, and Learned-Miller, Erik. Face detection with the Faster R-CNN. In *FG* (2017), IEEE.

[108] Jin, Aobo, Fu, Qiang, and Deng, Zhigang. Contour-based 3d modeling through joint embedding of shapes and contours. In *Symposium on Interactive 3D Graphics and Games* (2020), pp. 1–10.

[109] Kanazawa, Angjoo, Tulsiani, Shubham, Efros, Alexei A, and Malik, Jitendra. Learning category-specific mesh reconstruction from image collections. In *ECCV* (2018), pp. 371–386.

[110] Kanezaki, Asako, Matsushita, Yasuyuki, and Nishida, Yoshifumi. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *CVPR* (2018).

[111] Karras, Tero, Laine, Samuli, and Aila, Timo. A style-based generator architecture for generative adversarial networks. In *CVPR* (2019), pp. 4401–4410.

[112] Kingma, Diederik P, and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[113] Kingma, Diederik P, and Welling, Max. Auto-encoding variational Bayes. In *ICLR* (2014).

[114] Klare, Brendan F, Klein, Ben, Taborsky, Emma, Blanton, Austin, Cheney, Jordan, Allen, Kristen, Grother, Patrick, Mah, Alan, and Jain, Anil K. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *CVPR* (2015), pp. 1931–1939.

[115] Koestinger, Martin, Wohlhart, Paul, Roth, Peter M, and Bischof, Horst. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCVW* (2011).

[116] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. *NeurIPS 25* (2012).

[117] Kulhánek, Jonáš, Derner, Erik, Sattler, Torsten, and Babuška, Robert. ViewFormer: NeRF-free neural rendering from few images using transformers. In *ECCV* (2022).

[118] Laughlin, Andrew J., Sheldon, Daniel R., Winkler, David W., and Taylor, Caz M. Quantifying non-breeding season occupancy patterns and the timing and drivers of autumn migration for a migratory songbird using doppler radar. *Ecography 39*, 10 (10 2016), 1017–1024.

[119] Lee, Jaehoon, Bahri, Yasaman, Novak, Roman, Schoenholz, Sam, Pennington, Jeffrey, and Sohl-dickstein, Jascha. Deep Neural Networks as Gaussian Processes. *ICLR* (2018).

[120] Lempitsky, Victor, Kohli, Pushmeet, Rother, Carsten, and Sharp, Toby. Image segmentation with a bounding box prior. In *ICCV* (2009), IEEE, pp. 277–284.

[121] Lenc, Karel, and Vedaldi, Andrea. Understanding image representations by measuring their equivariance and equivalence. In *CVPR* (2015).

[122] Levin, Anat, Lischinski, Dani, and Weiss, Yair. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*. 2004, pp. 689–694.

[123] Levy, Axel, Matthews, Mark, Sela, Matan, Wetzstein, Gordon, and Lagun, Dmitry. MELON: Nerf with unposed images using equivalence class estimation. *arXiv:preprint* (2023).

[124] Li, Yin, Sun, Jian, Tang, Chi-Keung, and Shum, Heung-Yeung. Lazy snapping. *ACM TOG 23*, 3 (2004), 303–308.

[125] Liao, Shuai, Gavves, Efstratios, and Snoek, Cees GM. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *CVPR* (2019), pp. 9759–9767.

[126] Lin, Chen-Hsuan, Ma, Wei-Chiu, Torralba, Antonio, and Lucey, Simon. BARF: Bundle-adjusting neural radiance fields. In *ECCV* (2022).

[127] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *ECCV* (2014), Springer, pp. 740–755.

[128] Lin, Tsung-Yu, RoyChowdhury, Aruni, and Maji, Subhransu. Bilinear cnn models for fine-grained visual recognition. In *ICCV* (2015).

[129] Lin, Tsung-Yu, Winner, Kevin, Bernstein, Garrett, Mittal, Abhay, Dokter, Adriaan M., Horton, Kyle G., Nilsson, Cecilia, Van Doren, Benjamin M., Farnsworth, Andrew, La Sorte, Frank A., Maji, Subhransu, and Sheldon, Daniel. MistNet: Measuring historical bird migration in the US using archived weather radar data and convolutional neural networks. *Methods in Ecology and Evolution 10*, 11 (2019), 1908–1922.

[130] Lindenberger, Philipp, Sarlin, Paul-Edouard, Larsson, Viktor, and Pollefeys, Marc. Pixel-perfect structure-from-motion with featuremetric refinement. In *ICCV* (2021).

[131] Liu, Ming-Yu, and Tuzel, Oncel. Coupled generative adversarial networks. *NeurIPS 29* (2016), 469–477.

[132] Liu, Shaohui, Zhang, Yinda, Peng, Songyou, Shi, Boxin, Pollefeys, Marc, and Cui, Zhaopeng. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR* (2020), pp. 2019–2028.

[133] Liu, Shichen, Li, Tianye, Chen, Weikai, and Li, Hao. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *ICCV* (2019), pp. 7708–7717.

[134] Liu, Steven, Zhang, Xiuming, Zhang, Zhoutong, Zhang, Richard, Zhu, Jun-Yan, and Russell, Bryan. Editing conditional radiance fields. In *ICCV* (2021).

[135] Liu, Yachan, Perez, Gustavo, Cheng, Zezhou, Sun, Aaron, Hoover, Samuel, Fan, Wei, Maji, Subhransu, and Bai, Peng. Zeonet: 3d convolutional neural networks for predicting adsorption in nanoporous zeolites. *Journal of Materials Chemistry A* (2023).

[136] Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *ICCV* (2015).

[137] Lorenz, Dominik, Bereska, Leonard, Milbich, Timo, and Ommer, Bjorn. Unsupervised part-based disentangling of object shape and appearance. In *CVPR* (2019).

[138] Lowe, David G. Distinctive image features from scale-invariant keypoints. *IJCV* (2004).

[139] Lowe, David G. Distinctive image features from scale-invariant keypoints. *IJCV* (2004).

[140] Mahendran, Aravindh, and Vedaldi, Andrea. Visualizing deep convolutional neural networks using natural pre-images. *IJCV* (2016).

[141] Mahendran, Siddharth, Ali, Haider, and Vidal, René. 3d pose regression using convolutional neural networks. In *ICCVW* (2017), pp. 2174–2182.

[142] Mann, Henry B, and Wald, Abraham. On Stochastic Limit and Order Relationships. *The Annals of Mathematical Statistics 14*, 3 (1943), 217–226.

[143] Mariotti, Octave, Mac Aodha, Oisin, and Bilen, Hakan. Viewnet: Unsupervised viewpoint estimation from conditional generation. In *ICCV* (2021), pp. 10418–10428.

[144] Matthews, Alexander G de G, Rowland, Mark, Hron, Jiri, Turner, Richard E, and Ghahramani, Zoubin. Gaussian Process Behaviour in Wide Deep Neural Networks. *arXiv:1804.11271* (2018).

[145] Meng, Quan, Chen, Anpei, Luo, Haimin, Wu, Minye, Su, Hao, Xu, Lan, He, Xuming, and Yu, Jingyi. GNeRF: GAN-based Neural Radiance Field without Posed Camera. In *ICCV* (2021).

[146] Menon, Aditya, Van Rooyen, Brendan, Ong, Cheng Soon, and Williamson, Bob. Learning from corrupted binary labels via class-probability estimation. In *ICML* (2015), PMLR, pp. 125–134.

[147] Mescheder, Lars, Oechsle, Michael, Niemeyer, Michael, Nowozin, Sebastian, and Geiger, Andreas. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR* (2019), pp. 4460–4470.

[148] Meuleman, Andreas, Liu, Yu-Lun, Gao, Chen, Huang, Jia-Bin, Kim, Changil, Kim, Min H, and Kopf, Johannes. Progressively optimized local radiance fields for robust view synthesis. *arXiv preprint arXiv:2303.13791* (2023).

[149] Mildenhall, Ben, Srinivasan, Pratul P., Tancik, Matthew, Barron, Jonathan T., Ramamoorthi, Ravi, and Ng, Ren. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV* (2020).

[150] Mnih, Volodymyr, and Hinton, Geoffrey E. Learning to label aerial images from noisy data. In *ICML* (2012).

[151] Monnier, Tom, Fisher, Matthew, Efros, Alexei A., and Aubry, Mathieu. Share With Thy Neighbors: Single-View Reconstruction by Cross-Instance Consistency. In *ECCV* (2022).

[152] Müller, Thomas, Evans, Alex, Schied, Christoph, and Keller, Alexander. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG* (2022).

[153] Mur-Artal, Raúl, Montiel, J. M. M., and Tardós, Juan D. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* (2015).

[154] Murphy, Kieran, Esteves, Carlos, Jampani, Varun, Ramalingam, Srikumar, and Makadia, Ameesh. Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold. *arXiv preprint arXiv:2106.05965* (2021).

[155] Mustikovela, Siva Karthik, Jampani, Varun, Mello, Shalini De, Liu, Sifei, Iqbal, Umar, Rother, Carsten, and Kautz, Jan. Self-supervised viewpoint learning from image collections. In *CVPR* (2020), pp. 3971–3981.

[156] Neal, Radford M. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.

[157] Nettleton, David F, Orriols-Puig, Albert, and Fornells, Albert. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review* (2010).

[158] Niemeyer, Michael, Barron, Jonathan T., Mildenhall, Ben, Sajjadi, Mehdi S. M., Geiger, Andreas, and Radwan, Noha. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR* (2022).

[159] Noroozi, Mehdi, and Favaro, Paolo. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV* (2016).

[160] Novak, Roman, Xiao, Lechao, Bahri, Yasaman, Lee, Jaehoon, Yang, Greg, Hron, Jiri, Abolafia, Daniel A, Pennington, Jeffrey, and Sohl-Dickstein, Jascha. Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. In *ICLR* (2019).

[161] Novotny, David, Larlus, Diane, and Vedaldi, Andrea. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *CVPR* (2017).

[162] Novotny, David, Larlus, Diane, and Vedaldi, Andrea. Learning 3d object categories by looking around them. In *ICCV* (2017), pp. 5218–5227.

[163] Oord, Aaron van den, Li, Yazhe, and Vinyals, Oriol. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[164] Oquab, Maxime, Bottou, Léon, Laptev, Ivan, and Sivic, Josef. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR* (2015).

[165] Ozden, Kemal Egemen, Schindler, Konrad, and Gool, Luc Van. Multibody structure-from-motion in practice. *IEEE TPAMI* (2010).

[166] Ozuysal, Mustafa, Lepetit, Vincent, and Fua, Pascal. Pose estimation for category specific multiview object localization. In *CVPR* (2009), IEEE, pp. 778–785.

[167] Özyeşil, Onur, Voroninski, Vladislav, Basri, Ronen, and Singer, Amit. A survey of structure from motion. *Acta Numerica 26* (2017).

[168] Pan, Xingang, Zhan, Xiaohang, Dai, Bo, Lin, Dahua, Loy, Chen Change, and Luo, Ping. Exploiting deep generative prior for versatile image restoration and manipulation. In *ECCV* (2020), Springer, pp. 262–277.

[169] Papyan, Vardan, Romano, Yaniv, Elad, Michael, and Sulam, Jeremias. Convolutional Dictionary Learning via Local Processing. In *International Conference on Computer Vision* (2017), pp. 5306–5314.

[170] Park, Jeong Joon, Florence, Peter, Straub, Julian, Newcombe, Richard, and Lovegrove, Steven. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR* (2019), pp. 165–174.

[171] Pathak, Deepak, Krahenbuhl, Philipp, Donahue, Jeff, Darrell, Trevor, and Efros, Alexei A. Context encoders: Feature learning by inpainting. In *CVPR* (2016).

[172] Patrini, Giorgio, Rozza, Alessandro, Krishna Menon, Aditya, Nock, Richard, and Qu, Lizhen. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR* (2017), pp. 1944–1952.

[173] Pellacini, Fabio, Battaglia, Frank, Morley, R Keith, and Finkelstein, Adam. Lighting with paint. *ACM TOG 26*, 2 (2007), 9–es.

[174] Perez, Gustavo, Zhao, Wenlong, Cheng, Zezhou, Belotti, Maria, Deng, Yuting, Simons, Victoria, Tielens, Elske, Kelly, Jeffrey, Horton, Kyle, Maji, Subhransu, et al. Using spatio-temporal information in weather radar data to detect and track communal bird roosts. *bioRxiv* (2022).

[175] Pinheiro, Pedro O, Almahairi, Amjad, Benmaleck, Ryan Y, Golemo, Florian, and Courville, Aaron. Unsupervised learning of dense visual representations. *NeurIPS* (2020).

[176] Poggi, Matteo, Ramirez, Pierluigi Zama, Tosi, Fabio, Salti, Samuele, Mattoccia, Stefano, and Di Stefano, Luigi. Cross-spectral neural radiance fields. *arXiv preprint arXiv:2209.00648* (2022).

[177] Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[178] Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS* (2015).

[179] Ren, Xiaofeng. Finding people in archive films through tracking. In *CVPR* (2008).

[180] Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. Springer, pp. 234–241.

[181] Rosen, David M, Carlone, Luca, Bandeira, Afonso S, and Leonard, John J. SE-Sync: A certifiably correct algorithm for synchronization over the special euclidean group. *IJRR* (2019).

[182] Rosinol, Antoni, Leonard, John J, and Carlone, Luca. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641* (2022).

[183] Rother, Carsten, Kolmogorov, Vladimir, and Blake, Andrew. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM TOG 23*, 3 (2004), 309–314.

[184] Rublee, Ethan, Rabaud, Vincent, Konolige, Kurt, and Bradski, Gary. Orb: An efficient alternative to sift or surf. In *ICCV* (2011).

[185] Sagonas, Christos, Tzimiropoulos, Georgios, Zafeiriou, Stefanos, and Pantic, Maja. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCVW* (2013).

[186] Saha, Oindrila, Cheng, Zezhou, and Maji, Subhransu. Ganorcon: Are generative models useful for few-shot segmentation? *arXiv preprint arXiv:2112.00854* (2021).

[187] Saharia, Chitwan, Chan, William, Chang, Huiwen, Lee, Chris A, Ho, Jonathan, Salimans, Tim, Fleet, David J, and Norouzi, Mohammad. Palette: Image-to-image diffusion models. *arXiv preprint arXiv:2111.05826* (2021).

[188] Sajjadi, Mehdi S. M., Mahendran, Aravindh, Kipf, Thomas, Pot, Etienne, Duckworth, Daniel, Lučić, Mario, and Greff, Klaus. RUST: Latent Neural Scene Representations from Unposed Imagery. *CVPR* (2023).

[189] Sanchez, Enrique, and Tzimiropoulos, Georgios. Object landmark discovery through unsupervised adaptation. In *NeurIPS* (2019).

[190] Sarlin, Paul-Edouard, Cadena, Cesar, Siegwart, Roland, and Dymczyk, Marcin. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR* (2019).

[191] Sarlin, Paul-Edouard, DeTone, Daniel, Malisiewicz, Tomasz, and Rabinovich, Andrew. Superglue: Learning feature matching with graph neural networks. In *CVPR* (2020), pp. 4938–4947.

[192] Saxe, Andrew M., Koh, Pang Wei, Chen, Zhenghao, Bhand, Maneesh, Suresh, Bipin, and Ng, Andrew Y. On Random Weights and Unsupervised Feature Learning. In *ICML* (2011).

[193] Schmidt, Thorsten-Walther, Pellacini, Fabio, Nowrouzezahrai, Derek, Jarosz, Wojciech, and Dachsbacher, Carsten. State of the art in artistic editing of appearance, lighting and material. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 216–233.

[194] Schonberger, Johannes L, and Frahm, Jan-Michael. Structure-from-motion revisited. In *CVPR* (2016), pp. 4104–4113.

[195] Schönberger, Johannes L, Zheng, Enliang, Frahm, Jan-Michael, and Pollefeys, Marc. Pixelwise view selection for unstructured multi-view stereo. In *ECCV* (2016), Springer, pp. 501–518.

[196] Schönberger, Johannes Lutz, and Frahm, Jan-Michael. Structure-from-motion revisited. In *CVPR* (2016).

[197] Sedaghat, Nima, and Brox, Thomas. Unsupervised generation of a viewpoint annotated car dataset from videos. In *ICCV* (2015), pp. 1314–1322.

[198] Sermanet, Pierre, Frome, Andrea, and Real, Esteban. Attention for fine-grained categorization. *ICLRW* (2015).

[199] Shen, Yujun, Gu, Jinjin, Tang, Xiaoou, and Zhou, Bolei. Interpreting the latent space of gans for semantic face editing. In *CVPR* (2020), pp. 9243–9252.

[200] Shen, Yujun, Yang, Ceyuan, Tang, Xiaoou, and Zhou, Bolei. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE TPAMI* (2020).

[201] Shi, Yuge, Siddharth, Narayanaswamy, Paige, Brooks, and Torr, Philip HS. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *arXiv preprint arXiv:1911.03393* (2019).

[202] Shu, Zhixin, Sahasrabudhe, Mihir, Alp Guler, Riza, Samaras, Dimitris, Paragios, Nikos, and Kokkinos, Iasonas. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *ECCV* (2018).

[203] Simonyan, Karen, and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[204] Simonyan, Karen, and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[205] Sitzmann, Vincent, Zollhöfer, Michael, and Wetzstein, Gordon. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618* (2019).

[206] Smith, Cameron, Du, Yilun, Tewari, Ayush, and Sitzmann, Vincent. Flowcam:training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. *arXiv preprint arXiv:2306.00180* (2023).

[207] Sohl-Dickstein, Jascha, Weiss, Eric, Maheswaranathan, Niru, and Ganguli, Surya. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML* (2015), PMLR, pp. 2256–2265.

[208] Song, Xibin, Wang, Peng, Zhou, Dingfu, Zhu, Rui, Guan, Chenye, Dai, Yuchao, Su, Hao, Li, Hongdong, and Yang, Ruigang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *CVPR* (2019), pp. 5452–5462.

[209] Stepanian, Phillip M, Horton, Kyle G, Melnikov, Valery M, Zrnić, Dušan S, and Gauthreaux, Sidney A. Dual-polarization radar products for biological applications. *Ecosphere 7*, 11 (2016).

[210] Su, Hao, Qi, Charles R, Li, Yangyan, and Guibas, Leonidas J. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV* (2015), pp. 2686–2694.

[211] Su, Jong-Chyi, Cheng, Zezhou, and Maji, Subhransu. A Realistic Evaluation of Semi-supervised Learning for Fine-grained Classification. In *CVPR* (2021).

[212] Sucar, Edgar, Liu, Shikun, Ortiz, Joseph, and Davison, Andrew J. imap: Implicit mapping and positioning in real-time. In *ICCV* (2021).

[213] Sun, Jiaming, Shen, Zehong, Wang, Yuang, Bao, Hujun, and Zhou, Xiaowei. Loftr: Detector-free local feature matching with transformers. In *CVPR* (2021).

[214] Sun, Xingyuan, Wu, Jiajun, Zhang, Xiuming, Zhang, Zhoutong, Zhang, Chengkai, Xue, Tianfan, Tenenbaum, Joshua B, and Freeman, William T. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR* (2018), pp. 2974–2983.

[215] Suzuki, Masahiro, Nakayama, Kotaro, and Matsuo, Yutaka. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891* (2016).

[216] Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In *CVPR* (2015), pp. 1–9.

[217] Szeliski, Rick, and Kang, Sing Bing. Shape ambiguities in structure from motion. *IEEE TPAMI* (1997).

[218] Taketomi, Takafumi, Uchiyama, Hideaki, and Ikeda, Sei. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications* (2017).

[219] Tanaka, Daiki, Ikami, Daiki, Yamasaki, Toshihiko, and Aizawa, Kiyoharu. Joint optimization framework for learning with noisy labels. In *CVPR* (2018).

[220] Tatarchenko, Maxim, Richter, Stephan R, Ranftl, René, Li, Zhuwen, Koltun, Vladlen, and Brox, Thomas. What do single-view 3d reconstruction networks learn? In *CVPR* (2019), pp. 3405–3414.

[221] Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Yifan, W., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., Simon, T., Theobalt, C., Nießner, M., Barron, J. T., Wetzstein, G., Zollhöfer, M., and Golyanik, V. Advances in neural rendering. *CGF* (2022).

[222] Cheng, Zezhou, Chai, Menglei, Ren, Jian, Lee, Hsin-Ying, Olszewski, Kyle, Huang, Zeng, Maji, Subhransu, and Tulyakov, Sergey. Cross-Modal 3D Shape Generation and Manipulation. In *ECCV* (2022).

[223] Cheng, Zezhou, Esteves, Carlos, Jampani, Varun, Kar, Abhishek, Maji, Subhransu, and Makadia, Ameesh. Lu-nerf: Synchronizing local unposed nerfs for unsupervised pose estimation. In *ICCV* (2023).

[224] Cheng, Zezhou, Gabriel, Saadia, Bhambhani, Pankaj, Sheldon, Daniel, Maji, Subhransu, Laughlin, Andrew, and Winkler, David. Detecting and Tracking Communal Bird Roosts in Weather Radar Data. In *AAAI* (2020).

[225] Cheng, Zezhou, Gadelha, Matheus, and Maji, Subhransu. Accidental turntables: Learning 3d pose by watching objects turn. In *ICCVW* (2023).

[226] Cheng, Zezhou, Gadelha, Matheus, Maji, Subhransu, and Sheldon, Daniel. A Bayesian Perspective on the Deep Image Prior. In *CVPR* (2019).

[227] Cheng, Zezhou, Maji, Subhransu, and Sheldon, Daniel. AI for conservation: learning to track birds with radar. *XRDS: Crossroads, The ACM Magazine for Students* (2021).

[228] Therrien, Charles W. Issues in Multirate Statistical Signal Processing. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on* (2001), vol. 1, IEEE, pp. 573–576.

[229] Thewlis, James, Albanie, Samuel, Bilen, Hakan, and Vedaldi, Andrea. Unsupervised learning of landmarks by descriptor vector exchange. In *ICCV* (2019).

[230] Thewlis, James, Bilen, Hakan, and Vedaldi, Andrea. Unsupervised learning of object frames by dense equivariant image labelling. In *NeurIPS* (2017).

[231] Thewlis, James, Bilen, Hakan, and Vedaldi, Andrea. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV* (2017).

[232] Tian, Yonglong, Krishnan, Dilip, and Isola, Phillip. Contrastive multiview coding. *ECCV* (2020).

[233] Tishby, Naftali, Pereira, Fernando C, and Bialek, William. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[234] Tishby, Naftali, and Zaslavsky, Noga. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)* (2015).

[235] Tulsiani, Shubham, Efros, Alexei A., and Malik, Jitendra. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR* (2018).

[236] Tulsiani, Shubham, and Malik, Jitendra. Viewpoints and keypoints. In *CVPR* (2015), pp. 1510–1519.

[237] Ulyanov, Dmitry, Vedaldi, Andrea, and Lempitsky, Victor. Deep Image Prior. In *Computer Vision and Pattern Recognition (CVPR)* (2018).

[238] USGS, NLCD. Land cover (2011 edition, amended 2014), national geospatial data asset (ngda) land use land cover, 2011, editor. 2011. *US Geological Survey* (2011).

[239] Ustyuzhaninov, Ivan, Brendel, Wieland, Gatys, Leon A, and Bethge, Matthias. Texture Synthesis using Shallow Convolutional Networks with Random Filters. *arXiv:1606.00021* (2016).

[240] Van Horn, Grant, Mac Aodha, Oisin, Song, Yang, Cui, Yin, Sun, Chen, Shepard, Alex, Adam, Hartwig, Perona, Pietro, and Belongie, Serge. The iNaturalist species classification and detection dataset. In *CVPR* (2018).

[241] Van Rooyen, Brendan, Menon, Aditya, and Williamson, Robert C. Learning with symmetric label noise: The importance of being unhinged. In *NIPS* (2015).

[242] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[243] Wang, He, Sridhar, Srinath, Huang, Jingwei, Valentin, Julien, Song, Shuran, and Guibas, Leonidas J. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR* (2019), pp. 2642–2651.

[244] Wang, Nanyang, Zhang, Yinda, Li, Zhuwen, Fu, Yanwei, Liu, Wei, and Jiang, Yu-Gang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV* (2018), pp. 52–67.

[245] Wang, Yaxing, Gonzalez-Garcia, Abel, Berga, David, Herranz, Luis, Khan, Fahad Shahbaz, and Weijer, Joost van de. Minegan: effective knowledge transfer from gans to target domains with few images. In *CVPR* (2020), pp. 9332–9341.

[246] Wang, Zirui, Wu, Shangzhe, Xie, Weidi, Chen, Min, and Prisacariu, Victor Adrian. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064* (2021).

[247] Welling, Max, and Teh, Yee W. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *ICML* (2011).

[248] Wen, Bowen, Mitash, Chaitanya, Ren, Baozhang, and Bekris, Kostas E. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *IROS* (2020), IEEE, pp. 10367–10373.

[249] Wiles, Olivia, Koepke, A, and Zisserman, Andrew. Self-supervised learning of a facial attribute embedding from video. In *BMVC* (2018).

[250] Williams, Christopher K. I., and Rasmussen, Carl Edward. Gaussian Processes for Regression. In *NeurIPS* (1996), pp. 514–520.

[251] Williams, Christopher KI. Computing with Infinite Networks. In *NeurIPS* (1997).

[252] Winkler, David W. Roosts and migrations of swallows. *Hornero 21*, 2 (2006), 85–97.

[253] Winkler, David W., Hallinger, Kelly K., Ardia, Daniel R., Robertson, R. J., Stutchbury, B. J., and Cohen, R. R. Tree Swallow (Tachycineta bicolor), version 2.0. In *The Birds of North America*, P. G. Rodewald, Ed. Cornell Lab of Ornithology, 2011.

[254] Wu, Mike, and Goodman, Noah. Multimodal generative models for scalable weakly-supervised learning. *NeurIPS 31* (2018).

[255] Wu, Mike, and Goodman, Noah. Multimodal generative models for compositional representation learning. *arXiv preprint arXiv:1912.05075* (2019).

[256] Wu, Shangzhe, Rupprecht, Christian, and Vedaldi, Andrea. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR* (2020).

[257] Wu, Zhirong, Xiong, Yuanjun, Yu, Stella, and Lin, Dahua. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR* (2018).

[258] Xia, Yitong, Tang, Hao, Timofte, Radu, and Van Gool, Luc. Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. *arXiv preprint arXiv:2210.04553* (2022).

[259] Xiang, Yu, Kim, Wonhui, Chen, Wei, Ji, Jingwei, Choy, Christopher, Su, Hao, Mottaghi, Roozbeh, Guibas, Leonidas, and Savarese, Silvio. Objectnet3d: A large scale database for 3d object recognition. In *ECCV* (2016), Springer, pp. 160–176.

[260] Xiang, Yu, Mottaghi, Roozbeh, and Savarese, Silvio. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV* (2014), IEEE, pp. 75–82.

[261] Xiang, Yu, Schmidt, Tanner, Narayanan, Venkatraman, and Fox, Dieter. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199* (2017).

[262] Xiao, Shengtao, Feng, Jiashi, Xing, Junliang, Lai, Hanjiang, Yan, Shuicheng, and Kassim, Ashraf. Robust facial landmark detection via recurrent attentive-refinement networks. In *ECCV* (2016).

[263] Xiao, Tianjun, Xu, Yichong, Yang, Kuiyuan, Zhang, Jiaxing, Peng, Yuxin, and Zhang, Zheng. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR* (2015).

[264] Xiao, Tong, Xia, Tian, Yang, Yi, Huang, Chang, and Wang, Xiaogang. Learning from massive noisy labeled data for image classification. In *CVPR* (2015).

[265] Xiao, Yang, Du, Yuming, and Marlet, Renaud. Posecontrast: Class-agnostic object viewpoint estimation in the wild with pose-aware contrastive learning. In *2021 International Conference on 3D Vision (3DV)* (2021), IEEE, pp. 74–84.

[266] Xiao, Yang, Qiu, Xuchong, Langlois, Pierre-Alain, Aubry, Mathieu, and Marlet, Renaud. Pose from shape: Deep pose estimation for arbitrary 3d objects. *arXiv preprint arXiv:1906.05105* (2019).

[267] Xie, Yiheng, Takikawa, Towaki, Saito, Shunsuke, Litany, Or, Yan, Shiqin, Khan, Numair, Tombari, Federico, Tompkin, James, Sitzmann, Vincent, and Sridhar, Srinath. Neural fields in visual computing and beyond. *CGF* (2022).

[268] Xu, Jiarui, and Wang, Xiaolong. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *ICCV* (2021), pp. 10075–10085.

[269] Xu, Qiangeng, Wang, Weiyue, Ceylan, Duygu, Mech, Radomir, and Neumann, Ulrich. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711* (2019).

[270] Xu, Yinghao, Yang, Ceyuan, Liu, Ziwei, Dai, Bo, and Zhou, Bolei. Unsupervised landmark learning from unpaired data. *arXiv preprint arXiv:2007.01053* (2020).

[271] Yang, Gengshan, Sun, Deqing, Jampani, Varun, Vlasic, Daniel, Cole, Forrester, Chang, Huiwen, Ramanan, Deva, Freeman, William T, and Liu, Ce. Lasr: Learning articulated shape reconstruction from a monocular video. In *CVPR* (2021), pp. 15980–15989.

[272] Yang, Linjie, Luo, Ping, Change Loy, Chen, and Tang, Xiaoou. A large-scale car dataset for fine-grained categorization and verification. In *CVPR* (2015), pp. 3973–3981.

[273] Yang, Shuo, Luo, Ping, Loy, Chen-Change, and Tang, Xiaoou. Wider face: A face detection benchmark. In *CVPR* (2016), pp. 5525–5533.

[274] YM., Asano, C., Rupprecht, and A., Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *ICLR* (2020).

[275] Yu, Alex, Ye, Vickie, Tancik, Matthew, and Kanazawa, Angjoo. pixelNeRF: Neural radiance fields from one or few images. In *CVPR* (2021).

[276] Zadeh, Amir, Lim, Yao-Chong, Liang, Paul Pu, and Morency, Louis-Philippe. Variational auto-decoder: A method for neural generative modeling from incomplete data. *arXiv preprint arXiv:1903.00840* (2019).

[277] Zeiler, Matthew D, and Fergus, Rob. Visualizing and understanding convolutional networks. In *ECCV* (2014), Springer.

[278] Zhang, Jiahui, Zhan, Fangneng, Wu, Rongliang, Yu, Yingchen, Zhang, Wenqing, Song, Bai, Zhang, Xiaoqin, and Lu, Shijian. Vmrf: View matching neural radiance fields. In *ACM MM* (2022).

[279] Zhang, Richard, Isola, Phillip, and Efros, Alexei A. Colorful image colorization. In *ECCV* (2016).

[280] Zhang, Richard, Isola, Phillip, Efros, Alexei A, Shechtman, Eli, and Wang, Oliver. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR* (2018).

[281] Zhang, Richard, Isola, Phillip, Efros, Alexei A., Shechtman, Eli, and Wang, Oliver. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR* (June 2018).

[282] Zhang, Richard, Zhu, Jun-Yan, Isola, Phillip, Geng, Xinyang, Lin, Angela S, Yu, Tianhe, and Efros, Alexei A. Real-time user-guided image colorization with learned deep priors. *ACM TOG 9*, 4 (2017).

[283] Zhang, Song-Hai, Guo, Yuan-Chen, and Gu, Qing-Wen. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *CVPR* (2021), pp. 6012–6021.

[284] Zhang, Yuting, Guo, Yijie, Jin, Yixin, Luo, Yijun, He, Zhiyuan, and Lee, Honglak. Unsupervised discovery of object landmarks as structural representations. In *CVPR* (2018).

[285] Zhang, Yuxuan, Ling, Huan, Gao, Jun, Yin, Kangxue, Lafleche, Jean-Francois, Barriuso, Adela, Torralba, Antonio, and Fidler, Sanja. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR* (2021), pp. 10145–10155.

[286] Zhang, Zhanpeng, Luo, Ping, Loy, Chen Change, and Tang, Xiaoou. Facial landmark detection by deep multi-task learning. In *ECCV* (2014).

[287] Zhang, Zhanpeng, Luo, Ping, Loy, Chen Change, and Tang, Xiaoou. Learning deep representation for face alignment with auxiliary attributes. *IEEE TPAMI* (2015).

[288] Zhong, Yue, Gryaditskaya, Yulia, Zhang, Honggang, and Song, Yi-Zhe. Deep sketch-based modeling: Tips and tricks. In *3DV* (2020), pp. 543–552.

[289] Zhou, Bolei, Khosla, Aditya, Lapedriza, Agata, Oliva, Aude, and Torralba, Antonio. Learning deep features for discriminative localization. In *CVPR* (2016).

[290] Zhou, Yi, Barnes, Connelly, Lu, Jingwan, Yang, Jimei, and Li, Hao. On the continuity of rotation representations in neural networks. In *CVPR* (2019), pp. 5745–5753.

[291] Zhu, Jiapeng, Shen, Yujun, Zhao, Deli, and Zhou, Bolei. In-domain gan inversion for real image editing. In *ECCV* (2020), Springer, pp. 592–608.

[292] Zhu, Zihan, Peng, Songyou, Larsson, Viktor, Xu, Weiwei, Bao, Hujun, Cui, Zhaopeng, Oswald, Martin R, and Pollefeys, Marc. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR* (2022).

[293] Zhuang, Chengxu, Zhai, Alex Lin, and Yamins, Daniel. Local aggregation for unsupervised learning of visual embeddings. In *ICCV* (2019).