Mapping the World of Self-Supervised Learning

A Guide to Learning Representations from Unlabeled Data

This presentation breaks down the core principles, landmark methods, and future frontiers of SSL, a powerful paradigm that turns unlabeled data into a learning opportunity.

The Data Dilemma: The High Cost of Supervision

The Power and Problem of Supervised Learning

Supervised learning excels when given enough high-quality labeled data.

Problem: However, creating large, high-quality labeled datasets is incredibly expensive, time-consuming, and hard to scale. Think of the thousands of human hours required to label ImageNet.

HIGH COST OF MANUAL LABELING

The Untapped Potential of Unlabeled Data

In contrast, the world is awash in unlabeled data: trillions of images, all of YouTube's videos, digitized books, and the entire internet.

Opportunity: How can we leverage this massive resource to learn powerful, task-agnostic representations without manual labels?

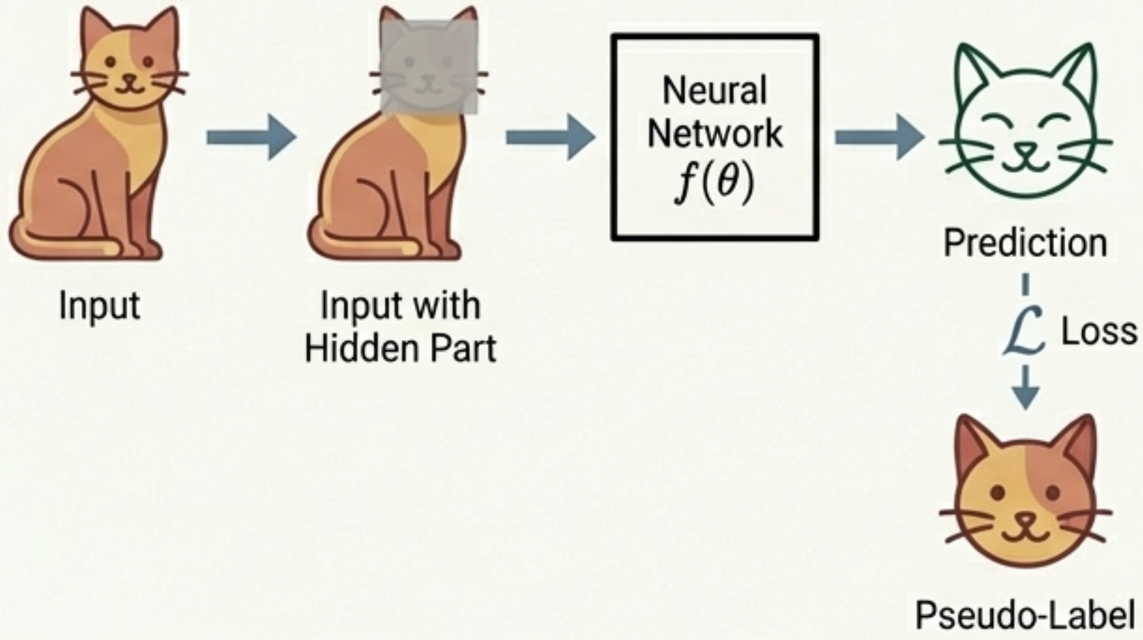
VAST OCEAN OF UNLABELED DATA RESOURCES

The Paradigm Shift: Self-Supervised Learning

> “Self-supervised learning... set us in a fairly smart way such that we can use supervised learning objective using only the unsupervised data.”

How it Works: The Pretext Task

- The core idea is to create a ‘pretext’ task from the unlabeled data itself.
- We hide some part of the input and train the model to predict it from the remaining parts.
- Because we know the ‘hidden’ part, we can generate our own ‘pseudo-labels’ for free.
- The model must learn rich semantic representations of the data to succeed at this task. These representations are the real prize.



The Power of Unsupervised Representations in Action

Example 1: Video Segmentation and Tracking

Source: Bondrik et al. (2018)

Pretext Task

Video Colorization. The model learns to predict the color of grayscale video frames.

Emergent Capability

To solve this, the model learns a rich representation of objects and their motion. This can be directly used for downstream tasks like object tracking and segmentation *without any fine-tuning.*

Frame 1: User Marks Object

Frames N, N+1, N+2: Model Propagates Mask

Example 2: Zero-Shot Image Classification with CLIP

Source: OpenAI's CLIP Model

Pretext Task

Contrastive learning on massive pairs of images and their associated text from the internet.

Emergent Capability

The model learns a shared embedding space for images and text, enabling powerful zero-shot classification on new tasks it has never seen.

✓ guacamole 98%
television studio
avocado
nachos
snack

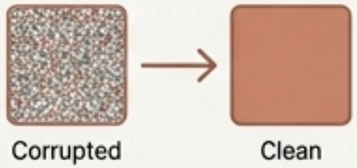
guacamole 98%
television studio
avocado
nachos
snack

Intellectual Foundations: The Precursors to Modern SSL

2006

Restricted Boltzmann Machines (RBMs) & Greedy Layer-wise Pre-training

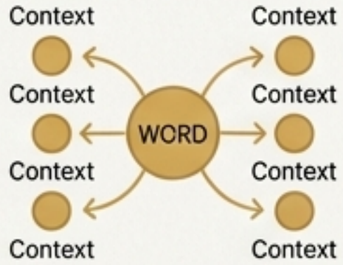
The original "unsupervised pre-training" paradigm that set the stage for training deep models.



2008

Denoising Autoencoders

Learned representations by reconstructing clean data from a corrupted version, a direct conceptual ancestor to masked prediction tasks like BERT.



~2013

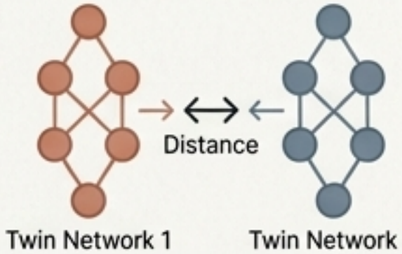
Word2Vec (Word Embeddings)

Models like Skip-gram and CBoW learned word vectors by predicting context, demonstrating the power of learning from local relationships.

2011

Autoregressive Models (NADE)

Predicted the next element in a sequence, a foundational technique for generative models like GPT.



1990s-2010s

Siamese Networks & Metric Learning

Trained twin networks to learn a metric space. Concepts like Triplet Loss and Contrastive Loss are the direct predecessors of modern contrastive learning.

Mapping the World of SSL: The Two Core Philosophies

Path 1: Self-Prediction

Learning from Within

Pretext tasks are defined *within a single data sample*. The model learns by predicting a hidden or corrupted part of the sample from the visible parts.

"Given this part of the data, what does the missing part look like?"

Self-Supervised Learning

Path 2: Contrastive Learning

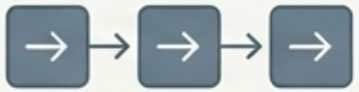
Learning by Comparison

Pretext tasks are defined by the *relationship between multiple samples*. The model learns to pull representations of "similar" samples together while pushing "dissimilar" ones apart.

"Which of these samples are similar to my anchor sample, and which are different?"

Path I: Self-Prediction – Learning from Within

Four Sub-categories of Self-Prediction Tasks



Autoregressive Generation

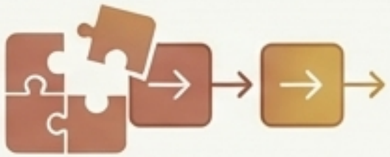
Predict the next element in a sequence, given the previous elements. Classic in domains with a natural order like language (GPT) and audio.

Mask Generation

Predict a randomly masked portion of the input from its surrounding context. Not dependent on a sequence order. The core idea behind BERT and denoising autoencoders.

Inner Relationship Prediction

Predict a property of the transformation applied to the data. Examples include predicting the relative position of shuffled image patches (Jigsaw) or the angle of rotation (RotNet).



Hybrid Approaches

Combine different generation models. For example, VQ-VAE learns a discrete codebook of visual parts, which a Transformer then models autoregressively.

Self-Prediction in Practice: Vision and Language

Vision Examples

Example 1: Jigsaw Puzzles

An image is divided into 9 patches and shuffled. The model must predict the correct permutation. To do this, it must learn about object parts and their spatial relationships.

Example 2: Rotation Prediction (RotNet)

An image is randomly rotated by 0, 90, 180, or 270 degrees. The model performs a 4-way classification to predict the rotation angle. This forces it to recognize the canonical orientation of objects.

0° 90° 180° 270°

Language Examples

Example 1: Masked Language Modeling (BERT)

Random words in a sentence are replaced with a `[MASK]` token. The model predicts the original words based on bidirectional context.

Example 2: Autoregressive Language Modeling (GPT)

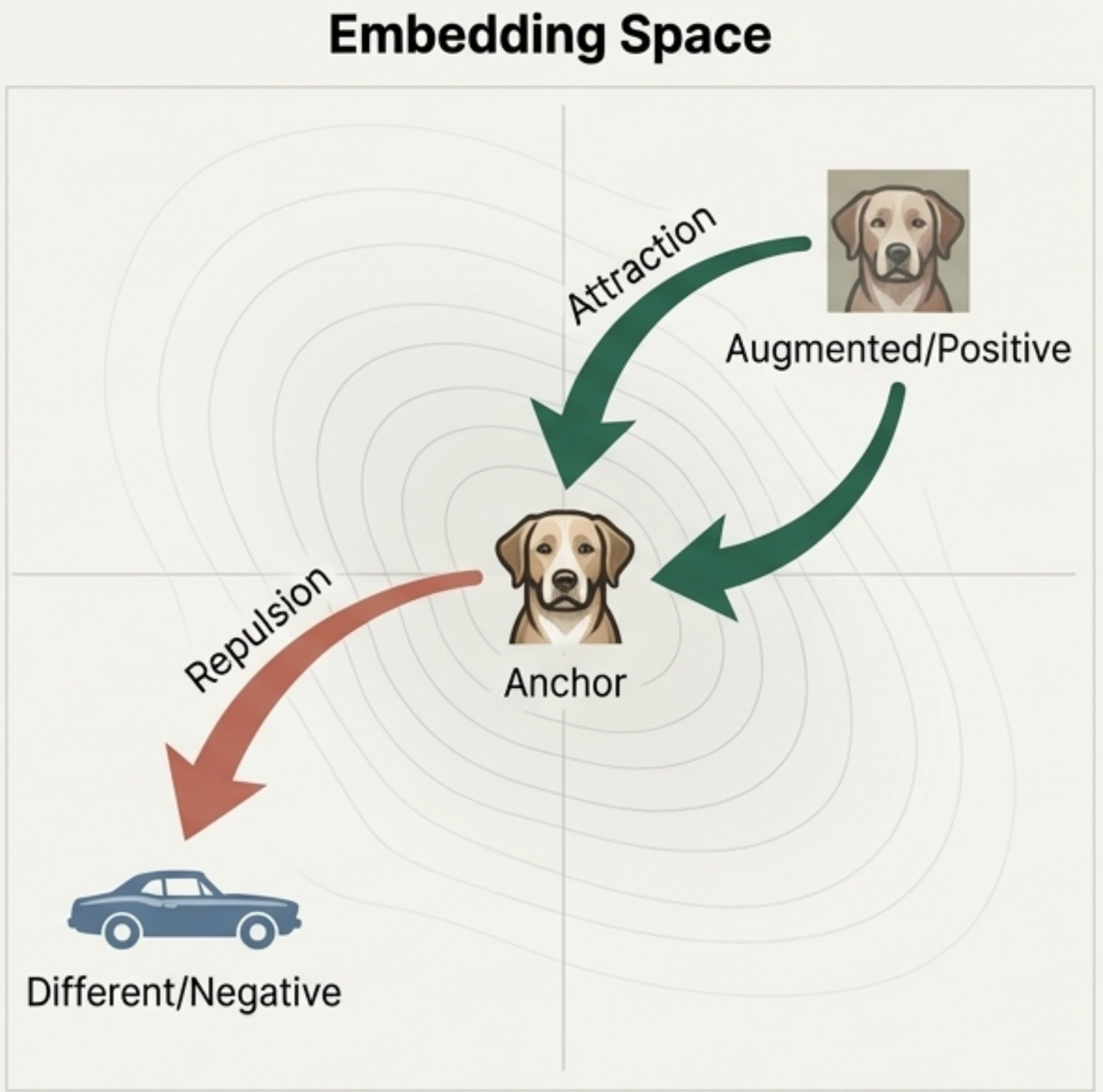
The model predicts the next word in a sequence, given all the preceding words.

Path II: Contrastive Learning – Learning by Comparison

Core goal: To learn an embedding space where similar sample pairs are close to each other, while dissimilar ones are far apart.

The “Anchor, Positive, Negative” Framework

- **Anchor:** The original data sample.
- **Positive:** A sample that should be “close” to the anchor. Typically an augmented view of the same anchor image.
- **Negative:** A sample that should be “far” from the anchor. Typically any other sample from the dataset.

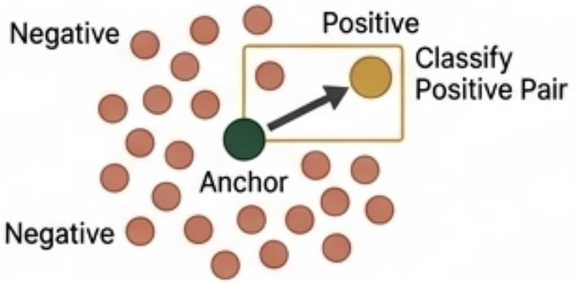


The Contrastive Learning Toolkit

Three Sub-categories of Contrastive Tasks

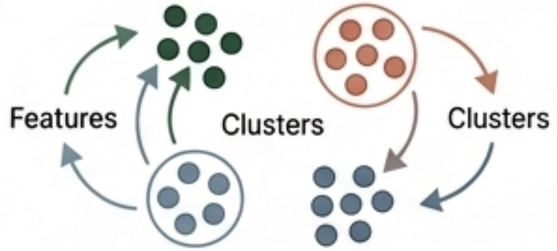
1. Category: Inter-Sample Classification (Most Dominant)

The task is framed as classifying which sample (out of many negatives) is the positive pair for a given anchor. This turns the problem into a classification task.



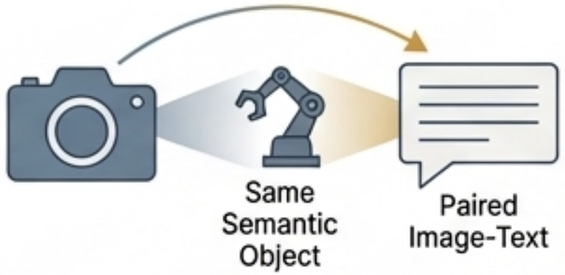
2. Category: Feature Clustering

Use an unsupervised clustering algorithm (like K-Means) to generate pseudo-labels. Then, samples within the same cluster are treated as positives. Thls creates a feedback loop: better features lead to better clusters, which leads to better features. (e.g., DeepCluster, SwAV).



3. Category: Multi-view Coding

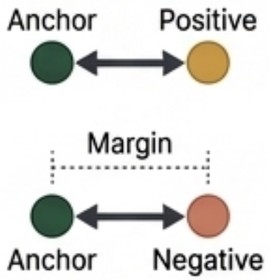
Positives are naturally occurring different "views" of the same semantic object. Examples include different camera angles of the same scene in robotics, or paired image-text data from the web (CLIP).



Key Loss Functions

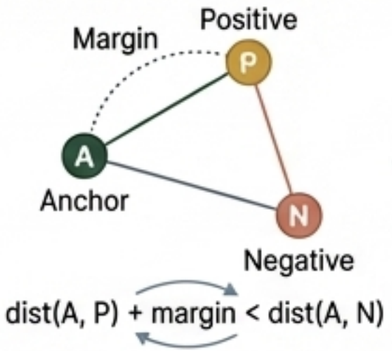
Loss 1: Contrastive Loss

Early loss that pulls positive pairs together if their distance is above a margin, and pushes negative pairs apart if their distance is below a margin.



Loss 2: Triplet Loss

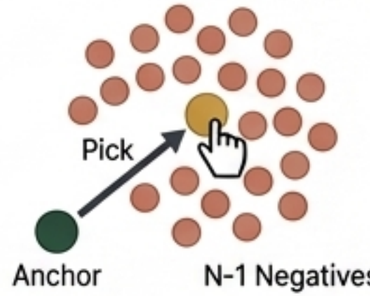
Generalizes this to an anchor, a positive, and a negative. Enforces that dist(anchor, positive) + margin < dist(anchor, negative)`



Loss 3: InfoNCE (Noise Contrastive Estimation)

The cornerstone of modern methods (used in CPC, MoCo, SimCLR).

Treats the task as identifying the single positive sample from a set of `N-1` negatives. Framed as a categorical cross-entropy loss.



Evolution of Landmark Contrastive Vision Models

InstDisc (Instance Discrimination)

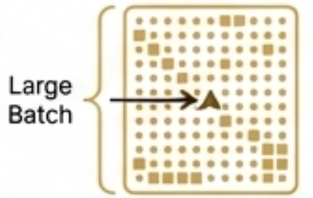
Core Idea: Treat each image instance in the dataset as its own class.
Innovation: Introduced a **memory bank** to store features for all instances, making million-way classification feasible.

MoCo (Momentum Contrast)

Core Idea: Builds on InstDisc but improves consistency.
Innovation: Replaced the static memory bank with a **queue** of recent batches.
Uses a **momentum encoder** (a slow-moving average of the main encoder) to provide consistent features for negative samples.

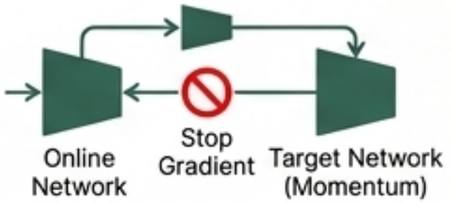
SimCLR (A Simple Framework for Contrastive Learning)

Core Idea: Drastically simplifies the framework.
Innovation: Showed that with sufficiently **large batch sizes**, you can get enough hard negatives directly from the batch, eliminating the need for a memory bank or momentum encoder. Relles heavily on strong data augmentation.



BYOL / SimSiam (Non-Contrastive Approaches)

Core Idea: Achieve state-of-the-art results *without explicit negative samples*.
Innovation: Use an asymmetric network (predictor head) and a stop-gradient operation to prevent collapse. BYOL uses a momentum encoder like MoCo. These models show that explicit repulsion might not be necessary if the architecture prevents trivial solutions.

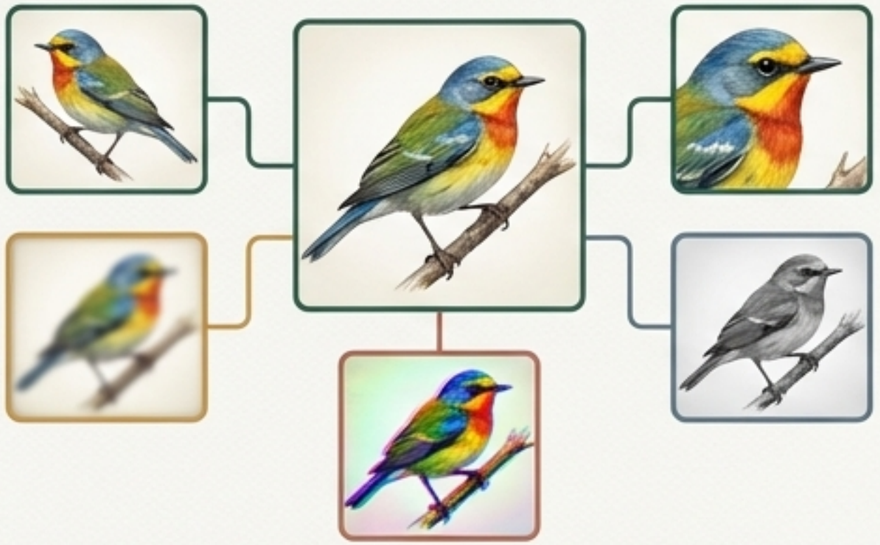


The Engine Room: Data Augmentation as a Source of Supervision

Why is Augmentation Critical?

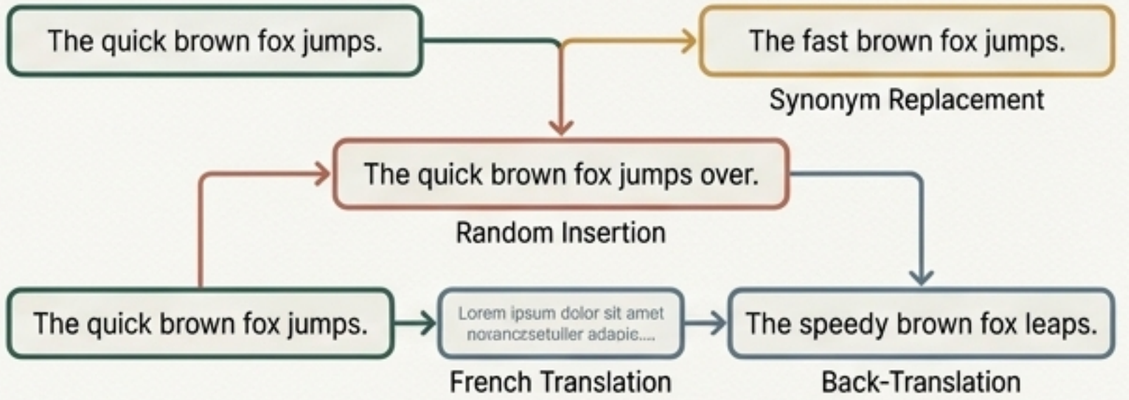
- In contrastive learning, augmentations create the “positive pairs” (different views of the same image).
- The choice of augmentations defines what invariances the model should learn. Strong augmentations force the model to focus on high-level semantic features rather than low-level shortcuts like color or texture.
- This is a form of injecting human prior knowledge about what makes an image “the same.”

Toolbox for Image Augmentation



Random Cropping & Resizing, Color Jittering & Distortion, Grayscale Conversion, Gaussian Blur, Random Horizontal Flip.

Toolbox for Text Augmentation

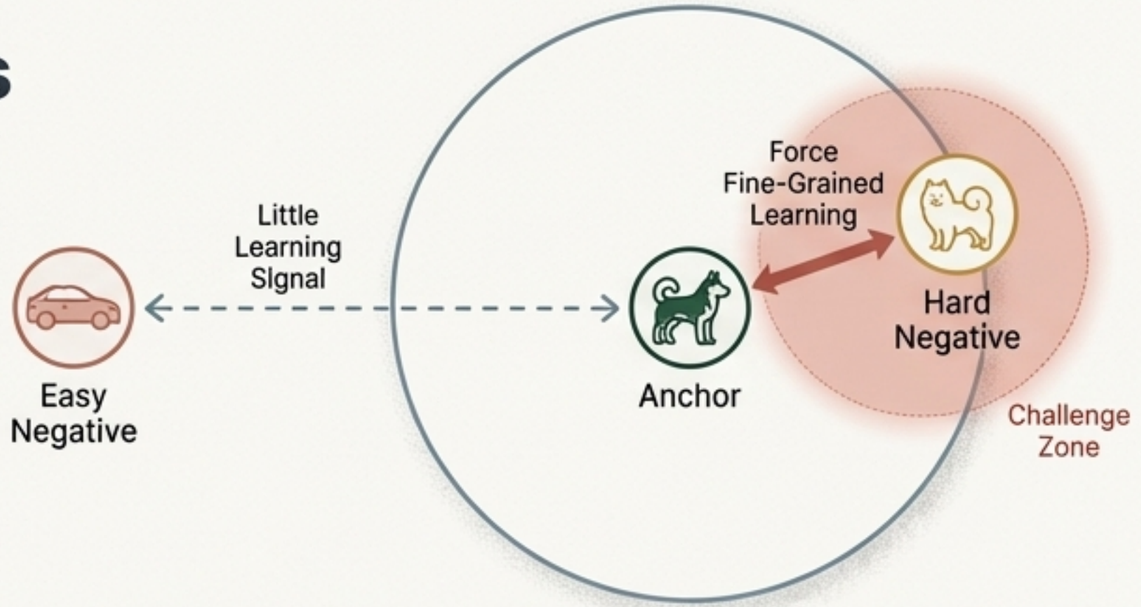


- **Lexical Edits:** Synonym Replacement, Random Insertion/Deletion.
- **Back-Translation:** Translate a sentence to another language and back to create a paraphrase.
- **Architectural Noise:** Using different dropout masks on the same input to create two distinct views (as in SimCSE).

Finding the Right Signal: The Importance of Hard Negatives

What is a Hard Negative?

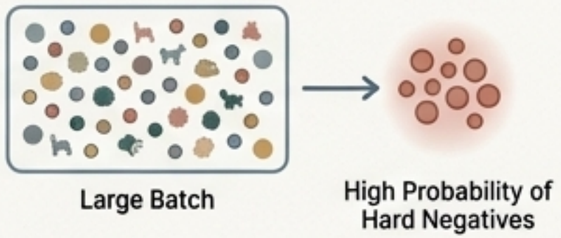
- A negative sample that the model finds difficult to distinguish from the anchor. In the embedding space, it is a dissimilar sample that lies very close to the anchor.
- **Why they matter:** Easy negatives (e.g., a cat vs. a car) provide little learning signal once the model is partially trained. Hard negatives (e.g., one breed of dog vs. another) force the model to learn fine-grained, distinguishing features.



Implicit Mining (The Modern Approach)

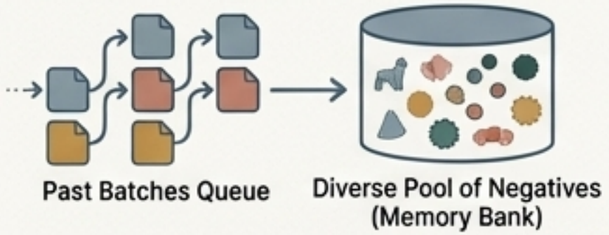
Large Batch Sizes

With thousands of samples in a batch (as in SimCLR), there is a high probability of naturally encountering hard negatives. This is computationally expensive but simple.



Memory Banks/Queues

By storing features from many past batches (as in MoCo), the model has a large and diverse pool of negatives to draw from.



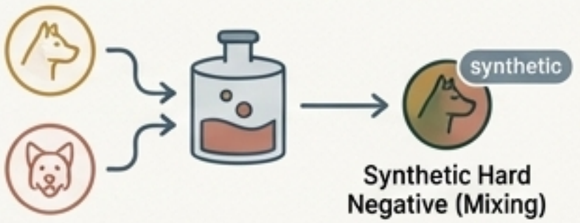
Explicit Mining

Pre-computing

Use an information retrieval system (like BM25) to find text samples with high keyword overlap but different meanings.

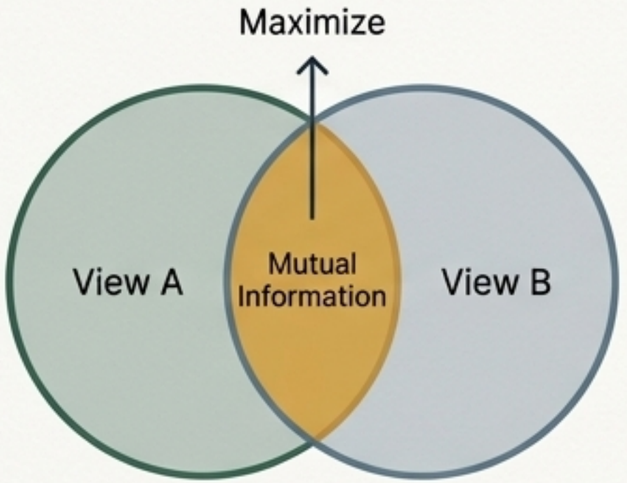
Mixing

Create new, synthetic hard negatives by mixing existing hard samples (e.g., MoCHi).



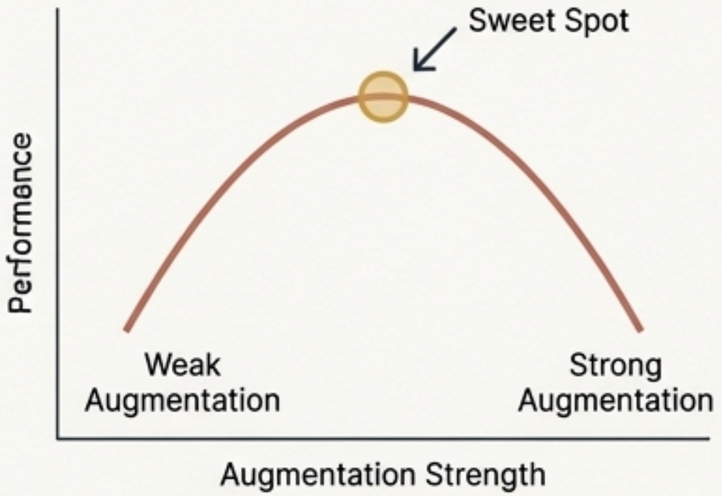
Theoretical Underpinnings: Why Does This Work?

The InfoMax Principle



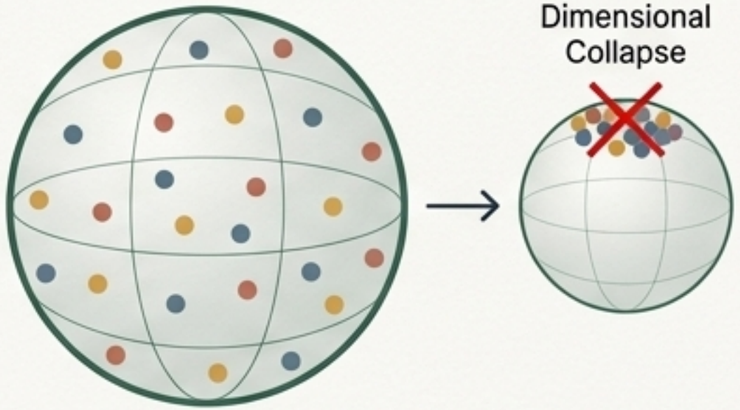
- The InfoNCE loss function is mathematically proven to be a lower bound on the **mutual information** between two views.
- By minimizing InfoNCE, the model is incentivized to maximize the information shared between the anchor and positive views, forcing it to learn the essential, invariant features.

The InfoMin Principle (The "Sweet Spot" for Augmentation)



- There's a tradeoff: if augmentations are too weak, views are too similar and the task is trivial. If they are too strong, they might destroy relevant information needed for downstream tasks.
- The goal is to find views that share just enough information to solve the downstream task (the "minimal sufficient" representation), and no more.

The Geometric View



- Contrastive learning has been shown to encourage a **uniform distribution** of features on the surface of a hypersphere.
- This "spreading out" of representations prevents dimensional collapse (where all features lie in a small subspace) and makes the features more useful for linear classifiers downstream.

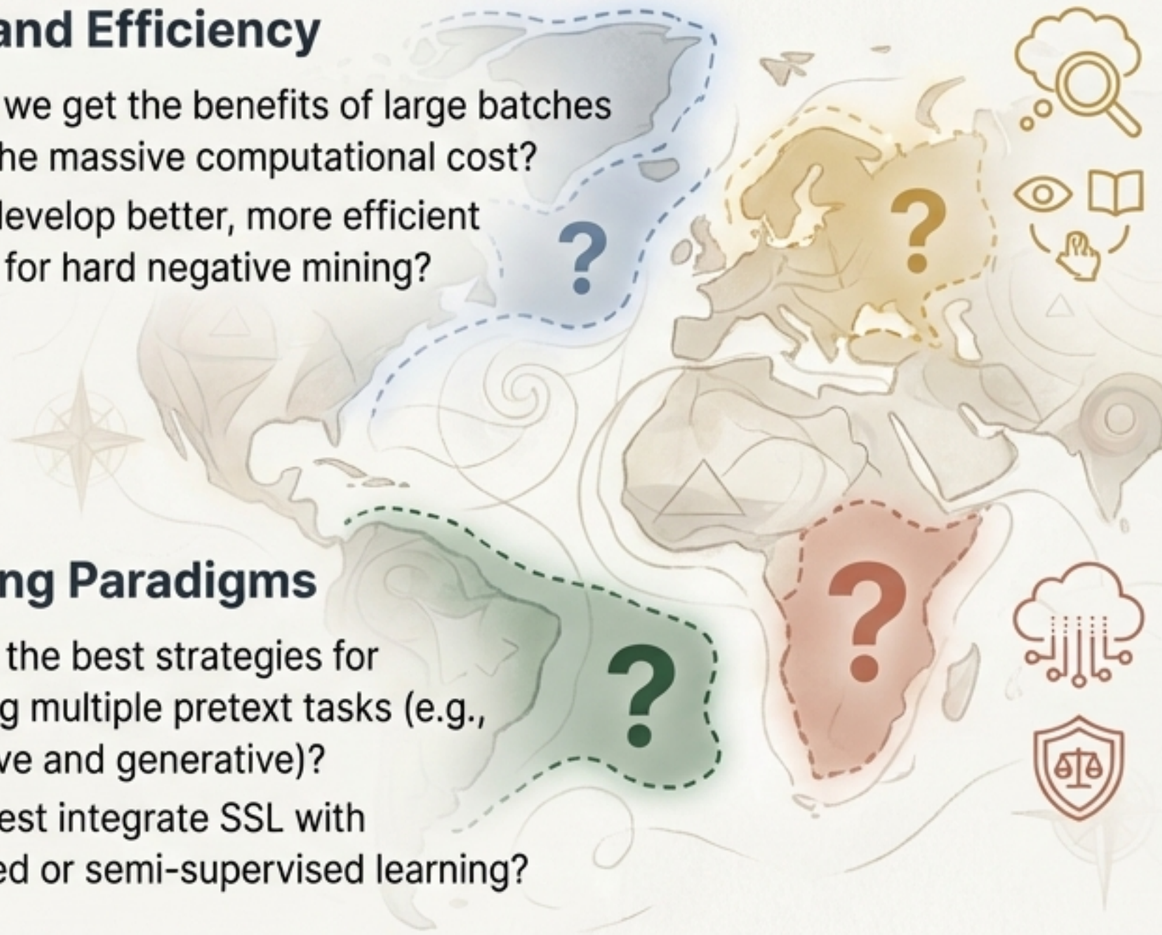
The Frontier: Open Questions and Future Directions

Scaling and Efficiency

- How can we get the benefits of large batches without the massive computational cost?
- Can we develop better, more efficient methods for hard negative mining?

Beyond Augmentation

- Developing a deeper theoretical understanding of why certain augmentations work.
- Creating universal augmentation strategies that aren't specific to one modality (e.g., vision, text).



Combining Paradigms

- What are the best strategies for combining multiple pretext tasks (e.g., contrastive and generative)?
- How to best integrate SSL with supervised or semi-supervised learning?

Data and Bias

- How to best leverage massive, noisy internet-scale datasets?
- Developing methods to measure and mitigate the social biases learned from unfiltered web data, especially in the learned embedding spaces.