A Review on Discriminative Self-supervised Learning Methods in Computer Vision

Nikolaos Giakoumoglou[1*], Tania Stathaki[1] and Athanasios Gkelias[1]

[1*]Department of Electrical and Electronic Engineering, Imperial College London, London, UK.

*Corresponding author(s). E-mail(s): nikos@imperial.ac.uk;
Contributing authors: t.stathaki@imperial.ac.uk;
a.gkelias@imperial.ac.uk;

Abstract

Self-supervised learning (SSL) has rapidly emerged as a transformative approach in computer vision, enabling the extraction of rich feature representations from vast amounts of unlabeled data and reducing reliance on costly manual annotations. This review presents a comprehensive analysis of discriminative SSL methods, which focus on learning representations by solving pretext tasks that do not require human labels. The paper systematically categorizes discriminative SSL approaches into five main groups: contrastive methods, clustering methods, self-distillation methods, knowledge distillation methods, and feature decorrelation methods. For each category, the review details the underlying principles, architectural components, loss functions, and representative algorithms, highlighting their unique mechanisms and contributions to the field. Extensive comparative evaluations are provided, including linear and semi-supervised protocols on standard benchmarks such as ImageNet, as well as transfer learning performance across diverse downstream tasks. The review also discusses theoretical foundations, scalability, efficiency, and practical challenges, such as computational demands and accessibility. By synthesizing recent advancements and identifying key trends, open challenges, and future research directions, this work serves as a valuable resource for researchers and practitioners aiming to leverage discriminative SSL for robust and generalizable computer vision models.

Keywords: Self-supervised learning, Computer Vision, Discriminative methods, Deep learning

1 Introduction

Over the past decade, artificial intelligence (AI) has made tremendous strides with the advent of deep learning, primarily through the development of convolutional neural networks (CNNs) [1] and, more recently, transformers [2, 3]. These architectures have become the dominant approaches for a large variety of tasks in computer vision, such as image classification, object detection, and image segmentation, and natural language processing (NLP), including sentence classification, language modeling, and machine translation. By using hierarchical feature extraction to efficiently process images, CNNs have long been the cornerstone of computer vision. However, the recent introduction of Vision Transformers (ViTs) [2, 3], by directly applying transformer architectures [4] –originally conceived for NLP– to image analysis, has led to substantial advancements in the field. ViTs allow the modeling of global dependencies and contextual information within images in a much more flexible and adaptive way for visual understanding. These deep learning models have been successful because they can discern complex patterns from the large datasets available today (i.e., CNNs excel at hierarchical feature learning, while ViTs use self-attention mechanisms to process whole images holistically), leading to state-of-the-art performance across various AI tasks.

The progress of AI systems has been fueled by large scale datasets that are carefully annotated by humans. While supervised learning has established itself as a robust method for training models on specific tasks, its application faces significant limitations, especially as we confront an era of data abundance. The rapid expansion of available data has outpaced our capacity to label it comprehensively. Deep learning models are particularly data-hungry, requiring vast amounts of labeled data to perform well [5]. Not exploring such large amounts of unlabeled data would not be only inefficient but also a waste of potential insights that could be gained from them. However, obtaining enough labeled data to train specialized models across various tasks can often be challenging and resource-intensive. In response to these challenges, the focus has shifted toward unsupervised learning approaches, particularly representation learning [6]. This shift is driven by the need to use the rich, unlabeled datasets by learning useful representations directly from the data itself. Representation learning aims to discover and learn the inherent structures and patterns within the data, often complex and subtle, without the need for explicit labeling. This approach is crucial in making AI models more adaptable and efficient, capable of handling a broader range of tasks with less reliance on expensive and labor-intensive data labeling processes.

Self-supervised learning (SSL) has rapidly emerged as a significant approach in AI, often referred to as *"the dark matter of intelligence"* [7]. This method diverges from traditional supervised learning, which relies on labeled datasets [8], by utilizing the abundant unlabeled data that are often underutilized, allowing machines to achieve a deeper understanding of the world without being limited by labeled datasets. SSL distinguishes itself from unsupervised learning by the way it leverages data. While both utilize unlabeled data to learn underlying representations, SSL is explicitly trained with supervisory signals that are generated from the data itself (self-supervision). This is achieved by leveraging the inherent structure of the data through various so-called pretext tasks [7]. These tasks, such as predicting the next frame in a video sequence [9]

or determining the jumbled order of a set of image patches [10], generate pseudo-labels which the model uses to learn features. This represents a fundamental shift towards learning versatile representations that can be applied across multiple tasks.

The concept of SSL originated in NLP, where it achieved significant success before being applied to computer vision. Models such as BERT [11, 12] and GPT [13] were among the first to show that self-supervised models could outperform supervised models, reaching top results in their field. These models are trained on large volumes of unlabeled text, showcasing the power of SSL to utilize unstructured data [14, 15]. Nevertheless, the success of these models can also be attributed to the distinct nature of language data, which is discrete and structured, compared to the continuous and unstructured nature of image data, making it a unique challenge for computer vision [16].

In computer vision, SSL has emerged as a method to extract robust features from unlabeled data using properties from the images themselves. It has overcome previous limitations regarding data size and quality, and models such as SEER [17, 18], trained on over of a billion images, illustrate the scale at which SSL can operate, pushing the boundaries of data handling and processing in unprecedented ways. The ability of SSL to match –and in some cases surpass– the performance of models trained on meticulously labeled data is a testament to its potential. Especially in competitive benchmarks like ImageNet ILSVRC-2012 [19, 20], SSL methods have demonstrated their prowess, challenging the traditional paradigms of deep learning. The wide applicability and effectiveness of SSL across various modalities underscore its significance as a cornerstone in the future development of AI technologies.

SSL has also been applied successfully in other areas such as video, audio, and time series analysis [9, 21–24]. These applications demonstrate its versatility in harnessing the intrinsic patterns of diverse data types to improve model performance and understanding.

SSL is generally categorized into two distinct approaches: discriminative and generative. The generative approach focuses on creating models capable of generating or reconstructing data to learn the overall data distribution. This method often employs models such as autoencoders (AEs) [25, 26] and generative adversarial networks (GANs) [27], which attempt to either reproduce the original data or generate new samples that are statistically similar to the input data. Despite its potential, generative SSL is noted for its high computational demands, particularly with high-resolution images, and questions remain about its utility and efficiency in learning meaningful representations [28, 29]. Conversely, discriminative SSL aims to develop good data representations by engaging in specific pretext tasks that don't require manual labels. These tasks are designed to enable the model to identify and differentiate between various features of the data, helping to enhance its capability to generalize these learned patterns to new, unseen data. This strategy evaluates the quality of the learned representations through a scoring function, akin to the methods used in traditional supervised learning [30].

Previous surveys on SSL have covered a wide range of aspects within this evolving field. However, many of these reviews tend to focus on particular areas of SSL or its specific applications within distinct domains [31–34]. One review highlights various

SSL methodologies specifically in computer vision [35], while another offers an overview focusing on pretext and downstream tasks along with some applications of SSL [36]. Another resource provides a comprehensive guide for training SSL methods, though it not formally a review article [7]. Additional reviews discuss SSL pretext tasks and architectures, offering a high-level overview of SSL applications in computer vision [37, 38]. Furthermore, a review that focuses on contrastive SSL methods also discusses the broad applications of SSL in both NLP and computer vision [39].

Several surveys have laid the groundwork for understanding SSL. One of the earliest and most comprehensive [38] provided an excellent initial overview of SSL techniques, covering image, video, and 3D vision. While a valuable initial resource, its coverage extends up to early 2020, preceding significant advancements in the field. Another important early work [37] focused specifically on contrastive SSL methods for both vision and natural language processing, offering a clear explanation of the contrastive learning pipeline. While these surveys were foundational, the rapid advancements in SSL, particularly after 2020, mean that their coverage of state-of-the-art techniques is now limited.

More recent surveys have attempted to capture the progress in the field. [39] offered a comprehensive review of contrastive learning methodologies but had a broader scope beyond computer vision, also including NLP, speech and text recognition and prediction. [40] provided a shorter survey focusing on comparing a limited number of only six discriminative SSL methods. [41] surveyed SSL methods for various domains, including computer vision, NLP, and graph learning, categorizing them by their objectives, i.e., generative, contrastive and adversarial. Similarly, [36] provided a general overview of the SSL pipeline and its applications across different domains, with less focus on detailed method analysis. [42] offered an in-depth look at the principles and components of contrastive learning but did not thoroughly analyze individual methods. [35] provided a historical view of image-oriented SSL, covering both generative and discriminative approaches. Finally, [7] presented a "cookbook" of SSL, outlining fundamental techniques with a unified vocabulary and highlighting key concepts like loss terms and training objectives. It further introduced practical implementation considerations, discussed common training recipes and evaluation methods, and shared practical insights from leading researchers. However, this paper serves more as a tutorial on the foundations and practical aspects of SSL rather than a comprehensive survey of methods.

In contrast to these broader surveys, several works have focused on specific application domains of SSL. [34] and [31] specifically addressed the application of SSL in medicine and healthcare. [33] concentrated on the use of SSL in medical imaging analysis. [32] explored SSL in remote sensing, considering different modalities. While these surveys provide valuable insights into domain-specific applications, our survey distinguishes itself by focusing specifically on discriminative SSL within the domain of computer vision. Furthermore, our work provides a more granular categorization of discriminative SSL methods, including contrastive, clustering, self-distillation, knowledge distillation, and feature decorrelation techniques. Significantly, our survey covers a substantially larger number of discriminative SSL methods and encompasses works

up to early 2025, offering a more up-to-date and focused perspective on this rapidly evolving area within computer vision.

The principal **contributions** of this review paper are summarized as follows:

- We provide a comprehensive introduction to the discriminative SSL pipeline, detailing its key components including pretext tasks, architectural choices, and loss objectives. This foundational material serves as essential background for understanding the subsequent in-depth review.
- We present a comprehensive and up-to-date review of discriminative SSL methods in computer vision, covering over 90 papers published from 2017 to 2025. We systematically categorize these methods into five primary groups: contrastive, clustering, self-distillation, knowledge distillation, and feature decorrelation, offering a thorough comparison of their pipeline and architectural designs.
- We detail the most commonly used evaluation protocols and datasets in discriminative SSL. We compile, analyze, and summarize a wide range of performance results reported across numerous methods using these standard protocols.
- We trace the architectural evolution of discriminative SSL methods, examining shifts in base networks, the progression of pretext tasks, and the development of training objectives and loss functions. We discuss how each of these advancements has contributed to the field's progress.
- We analyze the primary challenges faced by discriminative SSL, including an in-depth exploration of the inherent trade-offs within these methods.
- We discuss critical considerations regarding the evaluation and practical application of discriminative SSL techniques.

The rest of the review paper is structured as follows. Section 2 provides a background overview of SSL, introducing the pipeline, pretext tasks, downstream tasks, architectural components, and loss objectives essential for understanding the field. Section 3 comprehensively examines the five categories of discriminative SSL methods: contrastive, clustering, self-distillation, knowledge distillation, and feature decorrelation approaches. For each category, we analyze key representative methods, their architectural designs, and their unique contributions to the field. Section 4 presents a thorough evaluation framework, comparing method performance across standard benchmarks and transfer learning tasks. Section 5 offers critical insights into the evolution of architectural frameworks, fundamental challenges, inherent trade-offs, evaluation considerations, and theoretical foundations of discriminative SSL. Finally, Section 6 concludes the review by summarizing the key findings and suggesting promising directions for future research in this rapidly advancing field.

2 Background

The SSL pipeline typically consists of two main stages, the self-supervised pretraining and the supervised downstream task training stage, respectively, as depicted in Figure 1. In the pretraining stage, an unlabeled dataset is used to train a neural network (also referred to as the encoder, detailed in Section 2.3) on predefined pretext tasks (Section 2.1), where pseudo labels are generated automatically from certain data
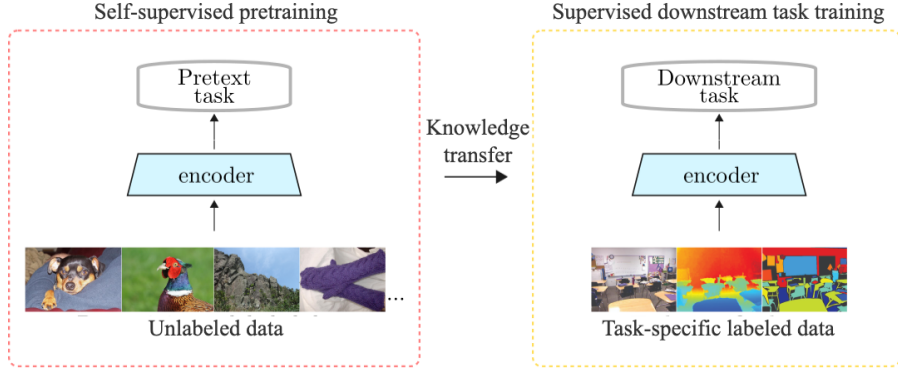
Fig. 1: Typical SSL pipeline. Once the training on the pretext task is complete, the learned parameters are adapted as a pre-trained model for various downstream computer vision tasks through fine-tuning. Image adjusted from [38].

attributes. These tasks are designed to encourage the encoder to learn useful representations from the data itself by developing filters that capture both basic and intricate features, which are beneficial for subsequent downstream tasks (Section 2.2), without the need for explicit labels. More specifically, the earlier layers of the encoder focus on general low-level features like corners, edges, and textures, while the deeper layers target more complex, high-level features such as objects, scenes, and parts of objects. The learning process is guided by carefully designed loss objectives (Section 2.4) that optimize feature discrimination, while data augmentation techniques (Section 2.5) generate diverse training views, enhancing robustness. Once the self-supervised training is complete, the learned visual features can be utilized as a pre-trained model for downstream tasks (Section 2.2), typically retaining only the initial layers of the encoder for fine-tuning with limited labeled data.

2.1 Pretext Tasks

A pretext task in SSL for computer vision is a carefully designed task that allows a model to learn useful visual representations from unlabeled data. These tasks leverage inherent properties of images to create pseudo-labels, eliminating the need for manual annotation. Early pretext tasks have included methods such as colorization (i.e., enhance the ability to understand textures, patterns, and object identities by colorizing grayscale images) [43], inpainting (i.e., use context prediction to fill in missing parts of an image) [44], jigsaw puzzles (i.e., learn to solve jigsaw puzzles, using the relative positions of image patches as supervision) [10, 45], geometric transformations (i.e., enhance feature learning for various orientations by predicting the rotation angle applied to an image) [46], and others.

More recent approaches have focused on contrastive learning via instance discrimination, a technique that trains models to distinguish between similar and dissimilar instances without using labeled data. Instance discrimination operates by maximising the similarity of positive pairs (typically augmented versions of the same image) while

minimizing the similarity of negative pairs (different images). Data augmentation, a key component of contrastive learning, involves applying various transformations to images, such as random cropping, resizing, colour distortion, or Gaussian blurring, to create different views of the same instance. In contrastive learning, data augmentation serves a dual purpose: it generates the positive pairs needed for the learning process and challenges the model to learn invariant representations across different transformations.

Data augmentation differs from conventional pretext tasks, which are specific, often handcrafted objectives designed to guide the model in learning useful representations by solving a predefined problem. While conventional pretext tasks focus on specific challenges, contrastive learning with data augmentation concentrates on recognizing different views of the same image, potentially leading to more general and transferable representations. For instance, as a pretext task, colorization involves training a model to predict colors for grayscale images as a learning objective. Besides, in data augmentation, color-related augmentations (e.g., color jittering and random grayscale conversions) are often used to create different views of the same image, serving the learning objective of contrastive learning: to distinguish between these augmented versions of the same image and different images.

Recently, Masked Image Modeling (MIM) was proposed as an effective pretext task for SSL in computer vision. MIM originated from NLP and gained prominence in computer vision with works like BEiT [47]. This task involves masking portions of an input image and training models to reconstruct the missing content, challenging them to understand visual context and structure. While effective across various architectures, including CNNs, MIM has shown particular success with ViTs. Unlike traditional pretext tasks, MIM focuses on reconstructing missing information rather than solving specific transformations. However, as a primarily generative method, MIM falls outside the scope of this review, which focuses on discriminative SSL techniques in computer vision.

Among the aforementioned pretext tasks, instance discrimination has become the dominant pretext task for discriminative SSL in computer vision and is widely adopted by most SSL methods. On the other hand, MIM is gaining ground as a powerful pretext task that enables generative SSL to match or even surpass instance discrimination.

2.2 Downstream Tasks

Downstream or target tasks are the primary objective of SSL, designed to leverage feature representations acquired during pretext training to tackle more complex problems, such as classification, object detection, semantic segmentation, and depth estimation. They directly assess the model's ability to understand and effectively utilize the learned features, frequently with limited labeled data, by employing techniques like fine-tuning or linear classification. Performance on these downstream tasks (e.g., image classification or object detection), serves as the primary measure of SSL success, quantifying the transferability and robustness of the learned representations and demonstrating the advantage of achieving high performance even with minimal supervision.
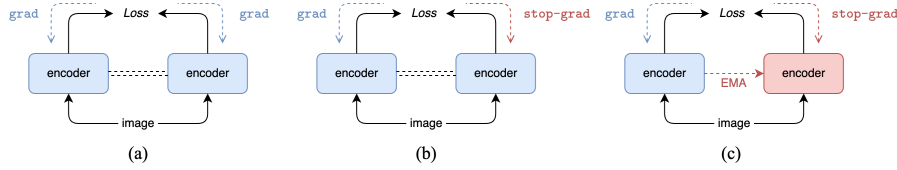
Fig. 2: Common architectural variations of Siamese networks used in self-supervised learning, each demonstrating different gradient flow mechanisms. (a) Siamese architecture with gradients (`grad`) flowing from both branches. (b) Siamese architecture where gradients flow from the first branch only. (c) Siamese architecture with gradient flow from the first branch and a stop-gradient (`stop-grad`) operation on the second branch, which is updated using an EMA.

2.3 Architectural Components

SSL uses different architectural components to effectively train models that can understand and represent complex visual data without labeled examples. The core of these architectures is the backbone (also referred to as the encoder), which acts as the primary feature extractor. This backbone transforms raw input images into a rich set of features. Initially, early SSL methods used CNNs like AlexNet [1] and VGG-16 [48], but more recent approaches often employ ResNets [49], their variants [50], or ViTs [2, 3, 51, 52].

Attached to the backbone is usually a projection network, which enhances the quality of representations produced by the encoder. While the backbone/encoder learns general features, the projector helps in optimizing the SSL objective without directly affecting the learned representations. This network is usually a multi-layer perceptron (MLP) that maps the high-dimensional features from the backbone into a space where similar examples are clustered together, aiding the learning of the images' semantic content. The projection network typically includes one to three layers and often features batch normalization to help stabilize the learning process. Some SSL models also include an additional prediction network, another MLP, to introduce asymmetry into the SSL architecture [29]. When transferring to downstream tasks, typically only the backbone is retained, while the projector is removed.

Modern SSL methods frequently employ dual backbone architectures, such as Siamese architectures (see Figure 2a), which consist of two identical neural networks that share the same weights. The fundamental premise of a Siamese architecture is to process two separate inputs simultaneously and compare their feature representations. This setup ensures consistent representations across different augmentations of the same image, improving the model's ability to generalize from unlabeled data.

In dual backbone architectures, the updating process can vary. Often, both backbones are updated through standard back-propagation. However, to prevent one branch of the Siamese network from affecting the learning of the other, a stop-gradient operation is sometimes employed [53] (see Figure 2b). This technique halts the gradient flow from one branch, effectively isolating the updates of each network and helping

to stabilize the training process by reducing interference between the branches and preventing the model from collapsing into trivial solutions.

Furthermore, in some cases, a momentum-based update method such as an exponential moving average (EMA) is employed for one of the backbones to help stabilize learning updates [16] (see Figure 2c). This approach can smooth out abrupt changes in model parameters over training iterations, potentially leading to faster convergence and better generalization, depending on the specific SSL method being used.

All these architectural components will be revisited in the remaining of this review when the different SSL techniques will be presented in detail.

2.4 Loss Objectives in Self-supervised Learning

In SSL, selecting appropriate loss objectives is crucial for effectively training models to process and interpret complex, unlabeled visual data. These loss functions are specifically designed to align with the goals of various SSL methods and have played a crucial role in the SSL evolution.

Cross entropy (CE) loss is commonly used in self-distillation and clustering methods to measure the difference between the predicted probabilities and the target distribution. It is particularly effective for models where outputs can be viewed as class probabilities. The formula for CE loss, when z_1 and z_2 are outputs from Siamese networks is given by:

$$\text{CE}(z_1, z_2) = -\sum_{i=1}^{N} z_{2,i} \cdot \log(z_{1,i}) \tag{1}$$

Mean squared error (MSE) loss and mean absolute error (MAE) loss are favored for regression tasks and methods that aim to closely match feature distributions. MSE calculates the average of the squares of the differences between predicted values and targets, ideal for embedding similarity, as shown by:

$$\text{MSE}(z_1, z_2) = \frac{1}{N} \sum_{i=1}^{N} (z_{11,i} - z_{21,i})^2 \tag{2}$$

MAE, offering robustness against outliers, computes the average magnitude of differences between pairs of observations:

$$\text{MAE}(z_1, z_2) = \frac{1}{N} \sum_{i=1}^{N} |z_{11,i} - z_{21,i}| \tag{3}$$

Negative cosine similarity (NCS) loss is prevalent in self-distillation, focusing on the angle between predictions rather than their magnitude. For ℓ_2-normalized vectors z_1 and z_2, NCS is calculated as:

$$\text{NCS}(z_1, z_2) = -\frac{z_1 \cdot z_2}{\|z_1\| \cdot \|z_2\|} \tag{4}$$

Kullback-Leibler (KL) divergence is utilized to measure the divergence of one probability distribution from another, which is crucial in predicting probability distributions. This measure is particularly useful in self-distillation methods that involve matching the output distributions of different network branches:

$$\text{KL}(P||Q) = \sum P \cdot \log \frac{P}{Q} \tag{5}$$

where P and Q are the true and predicted distributions, respectively. In this context, $P = softmax\,(z_2)$ and $Q = softmax\,(z_1)$.

A common approach in contrastive frameworks to extract meaningful representations from data without explicit labels, involves contrasting representations from augmented versions of the same data instance. The objective is to minimize the distance between these embeddings in the feature space, thus ensuring that the learned representations are invariant to the applied augmentations while being discriminative enough to distinguish among different instances. The initial loss function in this paradigm, known as contrastive loss, was initially introduced in [54, 55]. It aims to reduce the distance between positive pairs (i.e., different augmentations of the same instance) while enforcing a margin between negative pairs (i.e., augmentations of different instances). Building upon the contrastive loss, the triplet loss [56, 57] introduces a relative comparison between an anchor, a positive, and a negative sample, further refining the ability to capture differences between embeddings. To extend the comparison to multiple negative samples, the $(N+1)$-tuple loss [58] contrasts one positive pair against several negatives within the same batch. The Noise-Contrastive Estimation (NCE) loss [59] introduces a temperature parameter and employs explicit normalization to further refine the approximation of the probability distribution of the embeddings.

The InfoNCE loss [21], a variant of NCE, is widely adopted in SSL, particularly for contrastive learning models that leverage a large number of negative samples. It operates by maximizing the dot product similarity between positive pairs while minimizing similarity with negative samples.

$$\text{InfoNCE}(z_1, z_2) = -\log \frac{\exp(z_1 \cdot z_2/\tau)}{\sum_{k \neq 1} \exp(z_1 \cdot z_k/\tau)} \tag{6}$$

where z_1 and z_2 are augmented views of the same image, z_k represents other images in the batch, and τ is the temperature parameter. Subsequent developments have expanded upon the concepts introduced by InfoNCE [7]. [28] introduced the NT-Xent (Normalized Temperature-scaled Cross Entropy) loss, which diverges from the traditional InfoNCE loss by utilizing cosine similarity instead of the dot product:

$$\text{NT-Xent}(z_1, z_2) = -\log \frac{\exp(\text{sim}(z_1, z_2)/\tau)}{\sum_{k \neq 1} \exp(\text{sim}(z_1, z_k)/\tau)} \tag{7}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is a temperature parameter. As in InfoNCE, z_k represents the set of available negative samples, which can be drawn from a queue of previous samples [16] or from other samples within the current minibatch [28]. These negative samples are crucial as they serve as a reference point to enhance

the model's ability to recognize and differentiate between similar and different features by contrasting them against the positive pair.

Building upon the foundation laid by InfoNCE, subsequent research has explored various refinements [7]. The modification introduced in [60] excludes the positive pair from the InfoNCE denominator, thereby intensifying the differentiation between distinct instances. This adjustment sharpens the model's discriminative power, helping it to better isolate and identify unique features. Additionally, [61] enhanced the contrastive learning process by employing a queue to store embeddings of nearest neighbors. Moreover, [62] incorporated a regularization term to enforce invariance in the learned representations, thus adding an additional layer of constraint to the optimization problem. Finally, [63] utilized prototypes, i.e., representative embeddings of clusters, to guide the learning process and capture the underlying data structure.

2.5 Data Augmentation

Data augmentation plays a crucial role in SSL by generating diverse variations of input data, thereby enhancing model robustness and generalization. By applying various transformations to the original data while preserving semantic content, augmentation encourages the learning algorithm to identify consistent underlying patterns and instills representational consistency. This is essential for developing robust features applicable to downstream tasks. In SSL, data augmentation is instrumental in enriching the learning experience and improving the model's ability to generalize from its self-generated training signal.

Contrastive learning methods employing instance discrimination as the pretext task rely heavily on data augmentation to create distinct "views" of the same image. This approach is critical for training the model to recognize objects across different transformations. For example, by generating one augmented version of an image with color distortions and another with cropping or texture changes, the model learns to associate these disparate views as representing the same underlying object. The training objective is to bring the feature representations of these augmented views closer together in embedding space, while simultaneously distinguishing them from representations of different objects. This process significantly improves the model's ability to recognize objects under varying conditions, a key factor for strong performance on practical downstream tasks.

Key augmentation techniques include random resized cropping, random horizontal flipping, color jittering, random grayscale conversion, Gaussian blur, solarization, Sobel filtering, and multi-crop augmentation [64]. All these image manipulations aim to enhance the model's robustness to variations in orientation, color, and detail. The intensity and probability of each augmentation are controlled by specific parameters, offering a wide range of training variations. Figure 3 illustrates examples of these techniques. To optimize augmentation strategies, frameworks like AutoAugment [65] and RandAugment [66] have been developed. AutoAugment uses reinforcement learning to discover optimal augmentation policies for image classification, while RandAugment simplifies this process by controlling the magnitude of transformations with a single parameter, streamlining the identification of effective augmentations.
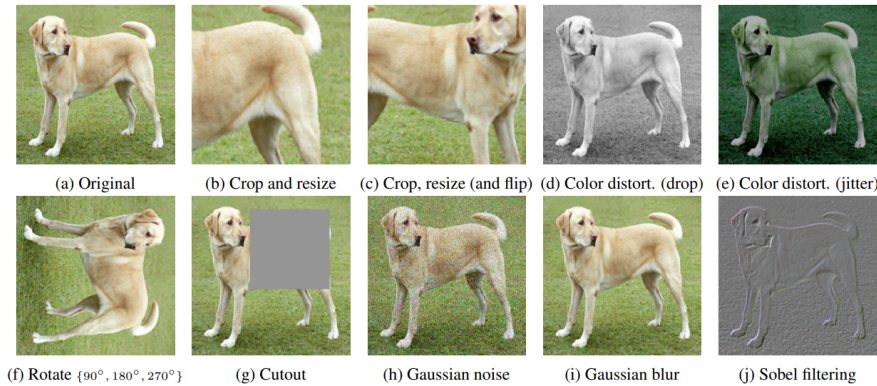
Fig. 3: Various data augmentation techniques applied to an original image of a yellow Labrador: (a) Original unmodified image; (b) Random crop and resize operation that focuses on a portion of the dog; (c) Similar crop and resize with possible horizontal flip; (d) Color distortion through grayscale conversion; (e) Color distortion through color jittering that alters the image's color properties; (f) Rotation applied at various angles (90°, 180°, 270°); (g) Cutout augmentation that masks a random square region of the image; (h) Gaussian noise injection that adds random pixel-level perturbations; (i) Gaussian blur that reduces image details while preserving structure; and (j) Sobel filtering that emphasizes edges and contours. These diverse transformations help self-supervised models learn invariant representations. Image from [28].

3 Self-supervised Learning Methods

This review categorizes discriminative SSL methods into five distinct yet overlapping categories, following the framework established in [7]. This classification provides a systematic basis for discussing how these techniques leverage unlabeled data to learn effective representations. While alternative classifications exist, we adopt this one to emphasize key attributes and mechanisms within each category.

Contrastive methods include techniques that learn representations by contrasting positive pairs against negative pairs. The key feature here is the use of a contrastive loss (e.g., InfoNCE or NT-Xent), which encourages similar samples (positives) to cluster closer in the embedding space while pushing dissimilar samples (negatives) further apart. These methods often rely on data augmentation to generate positive pairs and random sampling for negative pairs. The primary rationale for this categorization is the method's reliance on explicit pairing and negative sampling to drive the learning process, focusing on maximizing discriminative power between different instances. Contrastive learning methods will be discussed in detail in Section 3.2.

SimCLR [28] and MoCo [16] are arguably two of the most seminal models in contrastive SSL for computer vision. SimCLR's simplicity and strong performance, achieved through careful data augmentation and end-to-end training, established a powerful baseline. MoCo's innovative use of a momentum-updated encoder and a

dynamic queue of keys (negative samples) addressed the challenge of scaling contrastive learning, significantly improving performance and influencing subsequent work. Both models, which will be further analyzed later, have significantly advanced the field, often outperforming supervised learning approaches and bridging the gap between self-supervised and supervised learning in various computer vision tasks.

A key variant of contrastive learning is dense contrastive learning, which aims to learn dense feature representations essential for localization-based tasks like object detection and segmentation. These methods adapt the contrastive learning framework to address the unique challenges of these tasks. We dedicate a separate discussion to dense contrastive learning methods because their evaluation often differs from standard classification benchmarks, i.e., their objectives often lead to a trade-off that may reduce classification performance in favor of significant gains in object detection and segmentation. This distinct focus necessitates a dedicated analysis to highlight the unique challenges and solutions it presents.

Clustering methods utilise clustering algorithms to iteratively group and assign pseudo-labels to data points during training to supervise the learning process. This approach leverages the unsupervised clustering dynamics by promoting intra-cluster similarity and inter-cluster diversity to guide the representation learning without the need for negative samples. For example, DeepCluster [67] iteratively clusters features extracted from images and uses these cluster assignments to update the neural network, ensuring that the learned features are both diverse and informative. DeepCluster's specifics will be examined in Section 3.3.

Self-distillation methods involve training a single network to learn from its own predictions or representations, often at different layers or using different augmentations. Unlike contrastive methods, which rely on positive and negative sample pairs, self-distillation focuses on consistency between different views of the same input. This can be seen as a form of regularization, encouraging the network to learn more robust and generalizable features by distilling knowledge from its own outputs. This approach is especially beneficial for smaller models and can capture higher-level semantic information without explicit contrastive objectives. For instance, the network architecture of BYOL (Bootstrap Your Own Latent) [29], maximizes consistency between representations of two augmented views, focusing entirely on positive pair reinforcement and using techniques such as updating one encoder's weights with an EMA of the others to prevent feature collapse or using an asymmetrical architecture, introducing a predictor on the non-momentum branch. The detailed workings and effectiveness of BYOL will be explored in Section 3.4.

Knowledge distillation methods commonly utilize a teacher-student architecture, wherein a larger and more powerful pre-trained network (i.e., the teacher) distills its knowledge to a smaller network (i.e., the student). Knowledge distillation can be unsupervised, and when applied in an unsupervised manner, it can leverage SSL objectives to train the student network. This approach allows the student to learn from the richer representations captured by the teacher, often leading to improved performance, faster convergence, or the ability to train smaller, more efficient models. Essentially, the
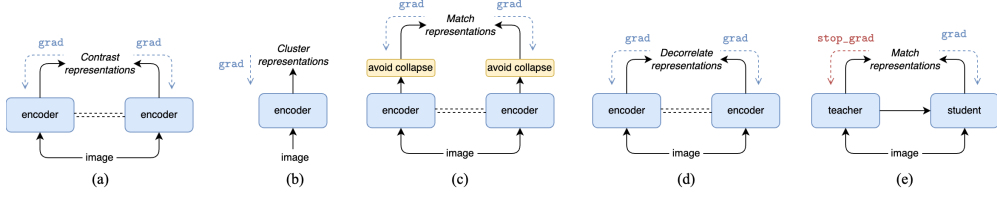
Fig. 4: High level comparison of discriminative SSL methods. (a) Contrastive methods employ instance discrimination. (b) Clustering methods use clustering algorithms to provide supervision. (c) Self-distillation methods use various techniques to avoid collapse while matching representations of Siamese networks. (d) Feature decorrelation methods promote feature diversity. (e) Knowledge distillation methods transfer knowledge from a frozen teacher network to a student network.

teacher guides the student, distilling its expertise and enhancing the student's ability to learn effective representations from unlabeled data. Knowledge distillation SSL methods will be discussed in detail in Section 3.5.

Feature decorrelation methods aim to learn representations where features are statistically independent or uncorrelated. This is often achieved by adding a decorrelation term to the loss function, encouraging the network to learn features that are diverse and non-redundant. By enforcing independence between features, they prevent dimensional collapse without relying on negative samples or asymmetric architectures. Unlike contrastive methods that focus on similarity and dissimilarity between different views, or self-distillation methods that emphasize internal consistency, feature decorrelation directly targets the statistical relationships. Feature decorrelation is exemplified by Barlow Twins [68], which enforces redundancy reduction across the features of its two neural networks by minimizing the deviation of the cross-correlation matrix from the identity matrix. This minimization enhances both discriminative ability and robustness. We analyse the implications and applications of Barlow Twins in Section 3.6.

Figure 4 provides a high-level overview, illustrating the key characteristics of the methods discussed above.

3.1 Early Works and Enablers

This section discusses early representation learning methods which, while initially proposed for diverse applications, have since been adapted or categorized within SSL frameworks. These foundational works introduced various pretext tasks designed to enable representation learning by exploiting different aspects of visual data. Techniques like image colorization, spatial transformations, and context prediction leveraged the inherent structure and content of images to learn useful features without labeled data. These approaches established a versatile foundation for SSL, demonstrating how creatively formulated tasks can drive the learning of rich visual representations.

Among these early methods, [43] focused on grayscale image colorization by predicting color channels from the lightness channel, addressing the multimodal nature of

colorization. Similarly, [69] developed a system that predicts per-pixel color histograms, effectively combining low-level and semantic representations and demonstrating the utility of colorization in self-supervised visual representation learning.

RotNet [46] trained a model to predict the rotation applied to an image, leveraging spatial transformations to learn robust image representations without labeled data. This method has significantly influenced subsequent work, establishing a foundational technique in SSL that exploits geometric transformations as a supervisory signal [70]. SplitBrain [71] utilized AEs to predict different color channels, focusing on cross-channel prediction within the network.

Context Encoders [44] employed a CNN trained to generate the contents of an arbitrary image region based on its surroundings, using a combination of reconstruction and adversarial loss. A related approach [30] trained a CNN to predict the relative position of image patches, again leveraging spatial context as a supervisory signal and requiring the recognition of objects and their parts.

Early work also explored training a CNN to solve jigsaw puzzles. Jigsaw [10] used a Context-Free Network (CFN) to learn about object parts and their spatial relationships. Subsequent work, Jigsaw++ [45], further improved performance by transferring knowledge, decoupling the self-supervised model's structure from the task-specific model, and leveraging pseudo-labels derived from clustered features of a pretext task.

Finally, adversarial learning saw significant progress with the introduction of Bidirectional Generative Adversarial Networks (BiGANs) [72], designed to learn the inverse mapping from data back to the latent space. While BiGANs are frequently employed in generative SSL frameworks, a discussion of these frameworks falls outside the scope of this paper due to their distinct generative objectives. Nevertheless, Limited Context Inpainting (LCI) [73], warrants mention. LCI aims to learn image representations by inpainting an image patch using a generator-discriminator architecture. Crucially, LCI preserves the patch boundary, maintaining local image statistics while altering global ones. The network is then trained to predict the type of transformation applied (rotation, LCI, or wrapping), as in [46]. These pioneering methods established a foundation for future SSL research and demonstrated the breadth of possible approaches and applications.

3.2 Contrastive Methods

Contrastive learning methods within deep metric learning commonly employ instance discrimination as a pretext task [7]. These methods revolve around three core components: an "anchor" sample (the reference point), a "positive" sample (similar to the anchor), and one or more "negative" samples (dissimilar to the anchor) [74]. The objective is to minimize the distance between the anchor and positive sample embeddings while maximizing the distance between the anchor and negative sample embeddings. In the absence of labels or other annotations, positive samples are often generated through augmentations of the anchor, while negative samples are typically randomly selected images from the mini-batch. Essentially, positive samples are created by generating multiple views of each data point. This can be achieved through various techniques, such as decomposing an image into its luminance and chrominance

components [75], applying random augmentations to an image multiple times [16, 28], extracting patches from the same image [21], or leveraging the outputs from student and teacher models [29]. Negative pairs are typically formed from randomly chosen images. Theoretically, positive pairs can be considered as samples from a joint distribution over views, $p(x_1, x_2)$, while negative pairs can be considered as samples from the product of marginals, $p(x_1) \cdot p(x_2)$ [76].

This section is organized into subsections based on the distinct characteristics and methodologies of each group of contrastive learning techniques. We begin by introducing foundational contrastive learning approaches (Section 3.2.1). We then explore Siamese network methods, which employ two parallel (often weight-sharing) networks and focus on learning through direct comparisons of different augmentations of the same image (Section 3.2.2). Next, we discuss momentum-based methods, which utilize a momentum encoder for gradual model updates, promoting stable learning on large, continually updated datasets (Section 3.2.3). Following this, we examine neighbourhood-based methods, which leverage data space proximity, frequently using nearest neighbours to improve the quality of learned representations (Section 3.2.4). We then present several frameworks that discuss comprehensive systems and architectures for more systematic implementation of contrastive learning (Section 3.2.5). Subsequently, we cover adversarial methods, which incorporate adversarial training to refine the contrastive learning process, typically by generating challenging negative samples (Section 3.2.6). Finally, we address dense contrastive methods, which focus on acquiring dense features for localization-based tasks (Section 3.2.8).

3.2.1 Foundational Approaches

Early SSL methods, such as InstDis [59] and PIRL [77], laid the groundwork for subsequent contrastive learning approaches. These initial methods demonstrated the potential of leveraging unlabeled data to learn robust representations by contrasting positive and negative sample pairs. In the following we summarize some of the most prominent foundational contrastive self-supervised methods.

Instance-level Discrimination (InstDis) [59] pioneered instance discrimination in SSL by introducing the idea of treating each image as its own unique class. The method learns by drawing different augmented views of the same image closer together in the embedding space, while pushing them away from all other images. Effectively, this creates a classification problem with as many classes as there are images in the dataset. InstDis trained a deep neural network using a non-parametric softmax classifier to map each image to a unit vector on a hypersphere. To address the computational challenges of this large-scale classification task, InstDis implemented a memory bank to store feature representations of all images, facilitating efficient negative example sampling during training. This method, by emphasizing the contrast between augmented views of the same image and a large pool of negative samples, established instance discrimination as a crucial pretext task and significantly influenced the development of later contrastive learning methods.

Unsupervised Embedding Learning (UEL) [78] aims to learn effective representations from unlabeled images by focusing on two key properties, i.e., invariance to data

augmentations and maximizing the spread between different instances. UEL employed a Siamese network with shared weights across two branches (each with a backbone and a projector) to process an original image and its augmented counterpart, along with a small batch of other images, generating ℓ_2-normalized embeddings. Framing the problem as binary classification, UEL uses maximum likelihood estimation to identify the augmented sample as belonging to the original image, distinguishing it from other instances in the batch. The objective is to minimize the negative log-likelihoods across all instances. UEL's key innovation is its instance-based softmax embedding approach, which directly optimizes instance features without relying on a memory bank. This emphasis on both invariance and spread-out influenced subsequent contrastive learning methods by demonstrating the importance of maintaining fine-grained visual consistency while effectively separating features from different instances.

Pretext-Invariant Representation Learning (PIRL) [77] learns image representations invariant to pretext transformations by encouraging similarity between representations of an original image and its transformed counterpart. It uses a single backbone network with two projection heads to learn representations invariant to image transformations, including the jigsaw pretext task. The method employs a memory bank to efficiently compare the current image with a large number of negative samples, allowing for effective contrastive learning without the need for large batch sizes [59]. PIRL's loss function incorporates two terms: the InfoNCE loss for positive pairs and its complement for differentiating positive from negative pairs. If the second term is disregarded, PIRL simplifies to an enhanced version of InstDis, referred to as InstDis++. By encouraging representations that are invariant under both data augmentations and pretext transformations, PIRL advanced contrastive learning by focusing on semantic consistency rather than task-specific features. This shift toward transformation-invariant representations directly influenced later contrastive frameworks, which also prioritize invariance properties.

Deep InfoMax (DIM) [79] and Augmented Multiscale Deep InfoMax (AMDIM) [80] learn representations by maximizing mutual information (MI) between input data and encoder outputs. DIM uses a contrastive objective to align global features (high-level summaries) with local features (intermediate activations), employing adversarial learning to match encoder outputs to a prior distribution. AMDIM extends DIM by maximizing MI across independently augmented views and multiple scales of the encoder's hierarchy, capturing richer cross-view and cross-scale information. DIM's key contribution was establishing MI maximization as a core SSL objective, formalizing contrastive learning as an information-theoretic problem and pioneering local-global hierarchical learning. AMDIM further advanced this by introducing multiview and multiscale contrast, enhancing robustness and capturing hierarchical semantic structures. These methods significantly influenced subsequent contrastive learning by emphasizing MI maximization and demonstrating the power of multiview and multiscale approaches, inspiring methods like CPC and CMC and providing a theoretical framework for understanding "good" representations.

Contrastive Predictive Coding (CPC) [21] learns representations by predicting future encodings in latent space. It uses an autoregressive model to generate predictions

of future representations based on a context vector, which is derived from past observations. The learning objective is to maximize the MI between the predicted future representations and the actual future observations, achieved through a contrastive loss (InfoNCE) that distinguishes true future samples from negative samples. CPC-v2 [81] enhances the original CPC framework by using a larger architecture, replacing batch normalization with layer normalization, and introducing improved data augmentation techniques. It also predicts patches from multiple directions, leading to more robust representations and improved performance on downstream tasks. By introducing the InfoNCE loss, CPC formalized contrastive learning as a probabilistic framework for maximizing MI. This loss function has proven to be highly effective in enhancing the learning of useful representations by maximizing the MI between related pairs of samples while minimizing it between unrelated ones. This approach influenced nearly all subsequent contrastive methods by highlighting the importance of temporal or sequential structure in representation learning.

Contrastive Multiview Coding (CMC) [75] explicitly addresses the multiview nature of data by leveraging different sensory modality or spatial views (e.g., luminance (L) and chrominance (ab colour channels)) to learn view-invariant representations. CMC trains a separate encoder for each view, maximizing mutual information between representations of different views of the same instance while minimizing it for different instances (as in InstDis [59]), effectively learning view-invariant features that capture shared information across multiple perspectives. This approach emphasizes the complementary nature of information from different views, a concept central to how contrastive learning uses augmentations as distinct "views" of the same data.

These foundational approaches, by exploring various contrastive strategies, laid the groundwork for modern SSL. Their focus on instance discrimination, mutual information maximization, and invariance to transformations paved the way for more sophisticated contrastive learning frameworks.

3.2.2 Siamese Networks Methods

Siamese networks, a neural network architecture using two or more identical subnetworks with shared weights to process different inputs and compare their outputs, have been introduced in early 1990s to solve signature verification as an image matching problem [54].

However, it was the Simple framework for Contrastive Learning of visual Representations (SimCLR) [28] that first adapted this architecture for self-supervised contrastive learning in computer vision, marking a significant milestone in the field. SimCLR introduced a remarkably simple yet highly effective framework that processes two augmented views of an image through identical networks, each consisting of a backbone and a projector, as shown in Figure 5a. The method generates these views through data augmentation and processes them through weight-sharing encoders to produce ℓ_2-normalized embeddings. SimCLR's learning objective focuses on maximizing the similarity between these embeddings using the NT-Xent loss, utilizing other batch samples as negative examples. SimCLR demonstrated that a simple contrastive approach, leveraging a Siamese architecture with shared weights to generate

contrastive pairs through appropriate data augmentation and large batch sizes, could outperform more complex methods. This work paved the way for subsequent SSL techniques like BYOL [29] and SimSiam [53] (formally introduced in Section 3.4), which further built upon this Siamese network foundation.

SimCLR-v2 [82] built upon the original SimCLR by incorporating larger ResNet models, a deeper projection head, and a memory bank from [16], along with integrating self-supervised pre-training with supervised fine-tuning.

G-SimCLR [83] offers another modification to the original SimCLR framework. It proposed a novel batch preparation strategy, beginning with training a denoising autoencoder [84] to obtain robust image representations. These representations are then clustered to assign pseudo-labels, and training batches are subsequently constructed to ensure that no pseudo-label is repeated within a batch.

Inspired by SimCLR, Simple Contrastive Learning (SimCL) [85] abandons data augmentation to create different views, and adopts adding uniform noise in the representation space. C-SimCLR [86] modifies SimCLR by adding information compression using conditional entropy bottleneck. Supervised Contrastive (SupCon) [74] extends the SimCLR framework to a fully supervised setting, efficiently leveraging label information. It introduces the concept of multiple positives per anchor, along with a broad set of negatives.

Inspired by SimCLR, DirectCLR [87] builds on the contrastive learning framework but distinguishes itself by eliminating the trainable projector typically employed in SimCLR to transform representations before applying the InfoNCE loss. Instead, DirectCLR directly optimizes a subvector of the representation space using the backbone encoder, retaining SimCLR's use of data augmentation to create different views. This approach prevents dimensional collapse—where embedding vectors span a lower-dimensional subspace—without relying on an additional projection layer.

3.2.3 Momentum-based Methods

Many of the aforementioned SSL methods face several critical challenges in representation learning. These include training instability caused by rapidly changing representations in methods like SimCLR, scalability limitations due to memory-intensive negative sampling requirements, and the risk of representation collapse where models output identical features for all inputs. Momentum techniques have emerged as a powerful solution to these issues.

Momentum contrast (MoCo) [16] was the first contrastive method to address these key challenges by introducing EMA to stabilize encoder updates, facilitating gradual weight adjustments that prevent abrupt shifts in representations. MoCo employs two encoders (as shown in Figure 5b): a query encoder (with parameters updated directly during training) consisting of a backbone and a projector, and a key encoder (with parameters updated using an EMA of the query network's parameters) with a similar architecture. This EMA update ensures that the keys in the dictionary are generated by a slowly evolving encoder, preventing the rapidly changing representations seen in other contrastive methods and maintaining representational consistency over time. Furthermore, MoCo utilizes a queue-based mechanism, rather than the memory bank

19

used in InstDis, to maintain a large and consistent dictionary of negative samples. This approach decouples the dictionary size from the mini-batch size, enabling the model to contrast against a significantly larger set of negative examples without increasing memory requirements. During training, the current mini-batch is encoded by the query encoder, while the keys in the queue are encoded by the momentum-updated key encoder, yielding ℓ_2-normalized embeddings. The contrastive loss (InfoNCE) is then computed between the query and the keys, where the positive key is the encoded augmented view of the same image, and the negative keys are drawn from the queue. Following each iteration, the oldest keys in the queue are replaced with the newly encoded keys from the current batch. This approach allows MoCo to efficiently leverage a large number of negative samples, enhancing the quality of learned representations while preserving computational efficiency.

MoCo-v2 [88] builds upon the original MoCo framework, incorporating several design improvements inspired by SimCLR [28]. While retaining MoCo's momentum-updated key encoder for stable representation learning and its dynamic queue to decouple batch size from the number of negatives (addressing SimCLR's reliance on large batch sizes), MoCo-v2 introduces several key changes. It replaces the single fully connected layer in the original MoCo with a 2-layer MLP projection head for more complex feature transformations. MoCo-v2 also adopts SimCLR's emphasis on stronger data augmentations, including blur and color distortion, to generate more diverse views of the same image and learn more robust representations. Furthermore, like SimCLR, MoCo-v2 uses ℓ_2-normalized embeddings and temperature scaling (τ) in its InfoNCE loss, focusing on directional similarity and stabilizing training for improved representation quality. These modifications have enabled MoCo-v2 to establish stronger baselines, outperforming SimCLR while using smaller batch sizes and fewer training epochs.

MoCo-v3 [89] builds upon its predecessors (MoCo and MoCo-v2) by introducing key architectural and methodological refinements tailored for ViTs and improving training stability. It maintains the Siamese network approach of the previous versions but simplifies the architecture by removing the memory queue, instead relying on in-batch negatives from larger batch sizes, and implements a symmetrized InfoNCE loss applied bidirectionally between query-key pairs. Moreover, it enhances stability, particularly for ViT training, by freezing the initial patch embedding layer or using fixed random projections to prevent gradient spikes, and adds an MLP prediction head to the query encoder for improved feature separation. Finally, it optimizes for ViTs by adopting the AdamW [90] optimizer, which is better suited for ViTs and enables stable large-batch training, along with a long warmup period of 40 epochs to mitigate early training instability. However, for ResNets it employs the LARS [91] optimizer. By addressing ViT-specific optimization challenges and refining contrastive learning mechanics, MoCo-v3 became a pivotal step toward modern SSL frameworks for vision. Table 10 summarizes the main differences between MoCo, MoCo-v2 and MoCo-v3 methods. The rest of this subsection presents several methods designed to address various limitations and performance issues associated with the MoCo framework.

Mixing of Contrastive Hard NegatIves (MoCHI) [92] addresses performance issues of MoCo by focusing on the challenge of "hard" negatives in contrastive learning.

MoCHI observes that as training progresses in MoCo, fewer negatives contribute significantly to the loss, indicating a need for harder negatives. To address this, it introduces a feature-level mixing strategy for hard negative samples, synthesizing new, more challenging negatives on-the-fly, leading to improved generalization and more efficient utilization of the embedding space. MoCHI demonstrates faster learning of transferable representations, showing higher performance gains over MoCo-v2 for transfer learning after only 100 epochs of training. However, it's worth noting that MoCHI does not show performance gains over MoCo-v2 for linear classification, suggesting its benefits may be task-specific or more pronounced in certain scenarios.

Building upon the observations of MoCHI and within the MoCo framework, SynCo [93] represents a significant advancement in contrastive learning through synthetic hard negative generation. While MoCHI introduced the concept of mixing hard negatives, SynCo substantially expands this approach by proposing four additional strategies beyond MoCHI's initial two methods. These strategies—extrapolated negatives, noise-injected negatives, perturbed negatives, and adversarial negatives—collectively create a more diverse and challenging set of synthetic negatives that target different aspects of the representation space. Unlike MoCHI, which showed limited gains in linear classification tasks, SynCo demonstrates consistent improvements across both linear evaluation and transfer learning benchmarks. This comprehensive approach to synthetic hard negative generation allows SynCo to modulate proxy task difficulty more effectively, facilitating faster training convergence and producing representations with better uniformity properties.

SimMoCo (Simple MoCo) and SimCo (SimMoCo with no MOmentum) [94] simplify the MoCo framework by removing the dictionary and momentum encoder, respectively, and integrating an InfoNCE loss with a dual temperature (DT) mechanism. This DT mechanism modulates the scaling of positive and negative sample distances within the loss function, enabling a more refined contrastive learning process where the separation between similar and dissimilar pairs can be finely tuned for enhanced discriminative feature learning. Their results demonstrate that neither the dictionary nor the momentum encoder is essential for training, proving that a simplified contrastive learning framework can still achieve robust performance by effectively leveraging the InfoNCE loss with this novel DT approach.

Information Minimization (InfoMin) [76] advances the MoCo framework by introducing a novel approach to data augmentation and contrastive learning. Building upon MoCo, InfoMin incorporates a specialized augmentation strategy called InfoMin Aug, which includes jigsaw augmentation alongside other techniques. The key innovation of InfoMin is its focus on minimizing MI between different views of the same image while maximizing the MI between representations and the underlying semantic content. In addition to jigsaw augmentation, InfoMin Aug. includes random resized crop, random horizontal flip, colour jitter, Gaussian Blur, RandAugment [66], and random grayscale. This comprehensive augmentation approach aims to create more challenging and informative positive pairs, encouraging the model to learn more robust and transferable representations.

Leave-one-out Contrastive (LooC) [95] builds upon the MoCo framework, introducing a novel approach to multi-augmentation contrastive learning. While MoCo uses

a queue of encoded samples as negative examples, LooC selectively leverages multiple augmentations of the same image. LooC's key innovation is treating one augmentation as the anchor while using the others as positive samples, effectively leaving out one augmentation at a time to create the anchor. Furthermore, it optimizes multiple augmentation-sensitive contrastive objectives using a multi-head architecture with a shared backbone, experimenting with rotation, color jittering, and texture randomization as augmentations. Unlike MoCo, which relies solely on different images for negative samples, LooC learns from subtle differences within augmentations of the same image, potentially improving its ability to distinguish fine-grained features.

Transformation Invariance and Covariance Contrast (TiCo) [96] advances MoCo by combining transformation invariance with covariance regularization. TiCo maximizes agreement among embeddings of different distorted versions of the same image, promoting transformation-invariant representations. To prevent trivial solutions, it regularizes the covariance matrix of embeddings from different images by penalizing low-rank solutions, effectively minimizing the distance between various augmentations of the same image and encouraging representations towards smaller eigenvalues of the covariance matrix, thus enhancing covariance contrast. Notably, TiCo can be viewed as a variant of MoCo with an implicit memory bank of unlimited size, but without additional memory cost, enabling it to outperform alternative methods with small batch sizes. By bridging contrastive and redundancy-reduction methods, TiCo provides new insights into joint embedding approaches and offers improved performance in computationally limited scenarios.

Relational Self-Supervised Learning (ReSSL) [97] advances MoCo by introducing a novel paradigm focused on maintaining relational consistency between instances under different augmentations, rather than explicitly pushing different instances apart as in MoCo. ReSSL models the relationship as a similarity distribution between a set of augmented images and uses this as a metric to align the same images with different augmentations. The method employs weak augmentations to represent more reliable relations, arguing that aggressive augmentations can hurt the reliability of target relations. ReSSL uses the KL divergence to measure and minimize the difference between the similarity distributions of two augmented views of the same image (see Figure 5c), effectively pushing for relational consistency. This approach, combined with a sharpened target distribution and a memory buffer with a momentum-updated network, allows ReSSL to achieve superior performance compared to MoCo-v2. Additionally, ReSSL demonstrates better performance on lightweight architectures, making it more suitable for smaller networks compared to traditional contrastive learning methods.

MoBY (MoCo-v2 with BYOL) [98] combines key elements from MoCo-v2 and BYOL [29] for SSL with ViTs (like DeiT-S [3] and Swin-T [51]). It adopts MoCo-v2's momentum design, key queue, and contrastive loss, while incorporating BYOL's asymmetric encoder architecture and data augmentations. This hybrid approach leverages a large negative sample dictionary and BYOL's ability to learn without explicit negatives, resulting in high accuracy on linear evaluation with ViT backbones. Notably, MoBY uses a smaller batch size and a small memory of size 4096 to achieve competitive performance.
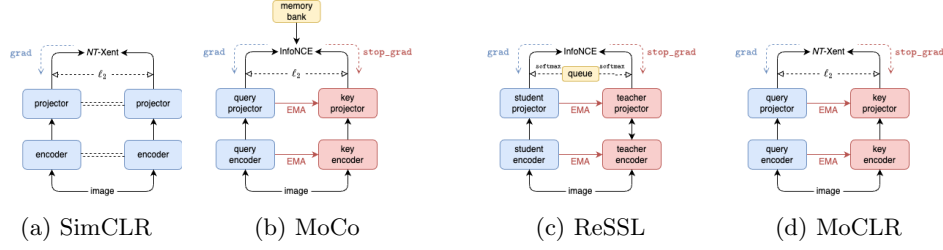
(a) SimCLR	(b) MoCo	(c) ReSSL	(d) MoCLR

Fig. 5: Overview of contrastive learning frameworks. (a) SimCLR uses simple data augmentations to generate positive pairs. (b) MoCo leverages a dynamic memory bank for negative samples. (c) ReSSL uses relation to look at relationships between instances. (d) MoCLR improves upon the SimCLR.

Momentum Contrastive Learning of Visual Representations (MoCLR) [99] combines key elements from MoCo, SimCLR and BYOL for enhanced SSL. It incorporates MoCo's momentum encoder and memory bank, SimCLR's multiple augmentations and non-linear projection head, and BYOL's prediction head and stop-gradient operations. MoCLR features two networks (shown in Figure 5d), each with a backbone and a projector, where the second network's parameters are updated via an EMA of the first network's parameters. Employing a symmetrized InfoNCE loss, MoCLR trains individual "expert" models on dataset subsets and distills their knowledge into a single model. Its unique dual contrastive learning mechanism applies loss between augmented views and between online and momentum networks, leveraging strengths from each method to improve representation learning and downstream task performance.

3.2.4 Neighborhood-based Methods

In contrastive learning, the terms "neighbor" or "neighborhood" typically refer to samples that are semantically or visually similar to a given anchor sample, often determined by proximity within the embedding space. This proximity is usually quantified using distance metrics like Euclidean distance or cosine similarity, which measure the closeness or separation of data points in the feature space. This subsection focuses on methods that leverage the concept of neighborhood to enhance contrastive learning by identifying and utilizing nearest neighbors.

Nearest-Neighbor Contrastive Learning of Visual Representations (NNCLR) [61] samples nearest neighbors from the dataset in the latent space and treats them as positive examples, providing more semantic variations than pre-defined transformations (see Figure 6a). It uses a support set, similar to MoCo's queue, to store embeddings representing the full data distribution. However, unlike MoCo which uses the queue for negative samples, NNCLR uses it to find positive pairs through nearest neighbor search. The method employs a contrastive loss similar to SimCLR's NT-Xent loss, but replaces one view of the positive pair with the nearest neighbor from the support set. This approach allows NNCLR to go beyond single-instance positives without resorting to clustering, leading to improved performance on various downstream tasks.

Mean Shift (MSF) [100] extends NNCLR by calculating the contrastive loss using the mean of the K most similar sample representations from a memory bank (see Figure 6c). This allows MSF to leverage multiple semantically similar samples, potentially providing a more robust and diverse set of positive examples.

All4One [101] advances NNCLR by introducing a novel contrastive SSL approach that combines three objectives: centroid contrast, neighbor contrast, and feature contrast. Drawing inspiration from other SSL methods, All4One incorporates elements from NNCLR for neighbor contrast, transformer-based models for the self-attention mechanism in centroid contrast, and redundancy reduction techniques like Barlow Twins (formally inroductd 3.6) for the feature contrast objective. The centroid contrast uses self-attention to create "centroids" from multiple neighbor representations, enabling the model to learn contextual information. The neighbor contrast, similar to NNCLR, facilitates learning directly from neighbors, while the feature contrast promotes learning unique feature representations.

Mine Your Own vieW (MYOW) [102], unlike NNCLR, employs a different mechanism to leverage a dataset's inherent diversity for richer representations. Building on BYOL's dual network framework and representation learning without negative pairs, MYOW mines the dataset itself for similar yet distinct samples to serve as views for one another, rather than relying solely on data augmentations. MYOW actively mines these views by identifying neighboring samples in the network's representation space and then predicting one sample's latent representation from a nearby sample. Specifically, it uses mined views in addition to augmented ones, employing a projector-predictor setup for augmented views and a projector-projector-predictor setup for mined views, with the objective of minimizing the NCS between them (Figure 6b). MYOW advances previous methods by overcoming limitations in domains where effective augmentations are unknown or limited, such as neuroscience, demonstrating the potential of harnessing data diversity for building rich representations where traditional augmentation techniques are insufficient.

MUlti-Granual Self Supervised learning (Mugs) [103][1] integrates elements from MoCo, MSF, All4One, and PCL, to boost performance in dense downstream tasks within a unified approach. Using a ViT architecture with momentum, Mugs combines instance discrimination with clustering via three distinct branches: instance-level, local-group, and group discrimination. The instance-level branch leverages a projector and predictor to implement the InfoNCE loss for direct instance comparison using a memory bank. The local-group branch combines patch tokens with their nearest neighbors through a bank using a transformer, aiming for localized context. Finally, the group discrimination branch abstracts to the cluster level using latent prototypes and a softmax-based CE loss.

3.2.5 Frameworks

SSL frameworks and methods serve distinct roles in the SSL landscape. SSL methods are specialized algorithms or techniques designed to address specific self-supervised

---

[1]Mugs is also a framework and a dense method capturing detailed (fine-grained) to very abstract (coarse-grained) features for downstream tasks.
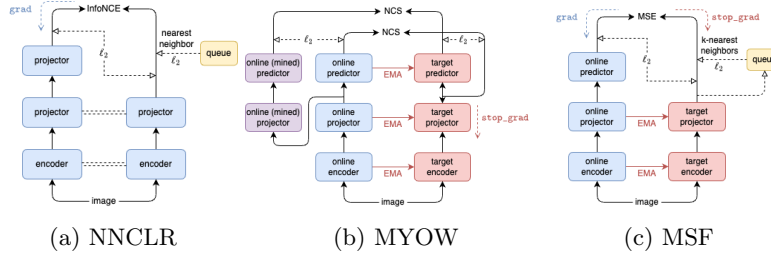
Fig. 6: Comparison of contrastive SSL architectures employing nearest neighbors. (a) NNCLR replaces the augmented view in SimCLR with its neighbour from a memory bank. (b) MYOW introduces a mined view to the BYOL framework. (c) MSF uses multiple nearest neighbours from a memory bank.

tasks, often tailoring their approaches to particular data types or applications. These methods implement unique combinations of components to achieve optimal performance in their target domains. In contrast, SSL frameworks provide comprehensive, flexible structures that encompass essential elements such as data augmentation strategies, pretext task definitions, and learning objectives. Frameworks offer a versatile foundation adaptable across diverse projects and domains, facilitating the development and comparison of multiple methods within a consistent environment. While methods focus on solving particular challenges with high efficiency, frameworks serve as the overarching architecture that enables the creation, implementation, and evaluation of various SSL approaches.

Yet Another DIM (YADIM) [104], shown in Figure 7b, is a contrastive SSL approach that builds upon the conceptual framework characterizing contrastive methods in five aspects: data augmentation pipeline, encoder selection, representation extraction, similarity measure, and loss function. YADIM merges the data processing pipelines of AMDIM and CPC to create a new variant that demonstrates competitive performance on various downstream tasks. It shows improved robustness to the choice of encoder and representation extraction strategy compared to other methods. YADIM is part of a broader implementation that includes standardized versions of several prominent SSL methods, namely AMDIM, CPC-v2, SimCLR, BYOL and MoCo-v2.

On the other hand, Decoupled Contrastive Learning (DCL) [60] framework provides a comprehensive structure for implementing contrastive learning by addressing the negative-positive-coupling (NPC) effect, which can diminish gradient effectiveness, particularly with uninformative samples. In addition, DCL introduces a specific method within this framework, modifying the widely used InfoNCE loss by removing the positive term from the denominator, improving learning efficiency and reducing sensitivity to batch size. This decoupling allows DCL to achieve competitive performance without requiring large batch sizes, momentum encoding, or extensive training epochs, demonstrating improved training efficiency and robustness to suboptimal hyperparameters compared to methods like SimCLR and MoCo-v2. Building on DCL's foundation, the variant DCLW introduces a weighting function for positive pairs, utilizing a negative

von Mises-Fisher distribution to adjust their importance. Both DCL and DCLW can be combined with any InfoNCE loss-based framework.

Representation Learning via Invariant Causal mechanisms (ReLIC) is primarily considered an SSL method rather than a framework. ReLIC introduces a novel approach to SSL by leveraging causal mechanisms to learn invariant representations. ReLIC works by training an encoder to produce representations that are invariant to data augmentations while being predictive of the original input. It uses a contrastive loss function that encourages similarity between augmented views of the same image and dissimilarity between different images. Specifically, it minimizes the InfoNCE loss and a regularization term implemented as the KL divergence among different views under different augmentation (Figure 7a). ReLIC's key innovation is the incorporation of causal principles, aiming to capture invariant features that are causally related to the input data rather than spurious correlations. Although ReLIC is generically defined, which is why it can be considered a framework that implements various methods including CPC, AMDIM, SimCLR, and BYOL, it is primarily utilized as a standalone SSL method in practice. ReLIC-v2 [105][2] is an extension of ReLIC that integrates an explicit invariance loss with a contrastive objective for dense downstream tasks. It incorporates a saliency mapping (images with background removed) within the augmentation and processes multiple data views of varying size.

Contrastive Learning with Stronger Augmentations (CLSA) is also primarily an SSL method rather than a framework. CLSA builds upon existing contrastive learning approaches by introducing stronger data augmentations to improve the quality of learned representations. CLSA works by applying more aggressive data augmentations to create diverse views of input images. These augmentations likely include techniques such as color jittering, random cropping, and geometric transformations. The method then uses a contrastive loss function to encourage similarity between augmented views of the same image while pushing apart representations of different images. By using stronger augmentations, CLSA aims to create more challenging positive pairs, forcing the model to learn more robust and generalizable features. While CLSA itself is a method, it incorporates elements from other SSL approaches, particularly those in the contrastive learning family. It likely builds upon methods like SimCLR and MoCo, as evidenced by its use of contrastive learning principles and data augmentation strategies.

Finally, UniGrad [106] is an SSL framework that proposes a unified approach by analyzing and unifying different SSL methods through gradient analysis. It derives a unified gradient formula encompassing various approaches, including contrastive learning methods (e.g., MoCo and SimCLR), asymmetric network methods (e.g., BYOL and SimSiam) that use `stop-grad` and predictors, and feature decorrelation methods (e.g., Barlow Twins and VICReg), as shown in Figure 7c. The framework demonstrates that these seemingly different methods can be unified into a similar gradient structure composed of positive and negative gradients. Based on this analysis, UniGrad proposes a concise gradient formula that explicitly maximizes the similarity between positive samples while expecting the similarity between negative samples to be zero. UniGrad

---

[2]ReLIC-v2 is also a dense method.
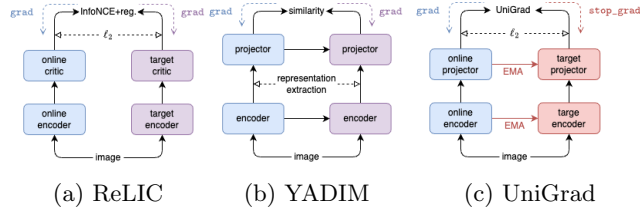
(a) ReLIC   (b) YADIM   (c) UniGrad

Fig. 7: Comparison of different contrastive SSL frameworks providing unified theoretical perspectives. (a) ReLIC incorporates causal mechanisms to learn invariant representations. (b) YADIM offers a systematic characterization of contrastive methods through standardized data processing pipelines and representation extraction strategies. (c) UniGrad presents a unified gradient analysis that connects seemingly disparate SSL approaches through their underlying gradient formulations.

does not explicitly include other SSL methods, but rather unifies and analyses them. It provides a framework for understanding and comparing different SSL approaches, showing that they work through similar mechanisms despite their apparent differences in loss functions. The UniGrad framework itself does not require a memory bank or a predictor network, making it a simple yet effective approach to SSL.

3.2.6 Adversarial Methods

Adversarial methods and data augmentation both aim to improve model performance, but differ significantly. Augmentation uses predefined, computationally inexpensive transformations to increase data diversity and prevent overfitting, often resulting in visibly altered images. Adversarial methods, conversely, dynamically generate challenging, often imperceptible examples through computationally demanding optimization processes, leveraging gradient information to adapt to the model's current state and target specific weaknesses. While augmentations are generally model-agnostic and broadly applicable, adversarial methods can be tailored to specific model weaknesses, creating increasingly difficult examples as the model improves. In SSL, adversarial contrastive learning employs several techniques to enhance robustness and feature learning. These include generating instance-wise adversarial examples by adding small perturbations to maximize contrastive loss, producing harder negative pairs within a batch through adversarial optimization, and combining both approaches by creating challenging positive pairs through adversarial augmentations and harder negative pairs through adversarial training. Furthermore, techniques like multi-layer and cluster-driven perturbations generate adversarial examples based on features from different network layers or the representation space's clustering structure. Ultimately, while augmentation indirectly enhances performance through increased data diversity, adversarial methods in SSL directly challenge the model's understanding, fostering robustness and more discriminative feature learning by exposing the model to more challenging scenarios.

As an early adversarial SSL method, Contrastive Learning with Adversarial Examples (CLAE) [107] enhanced contrastive learning by incorporating adversarial examples. CLAE generates perturbations to augmented images, creating harder positive pairs for the contrastive loss. This forces the model to learn features invariant to these adversarial changes, resulting in more robust and discriminative representations. Influenced by contrastive learning methods like SimCLR and MoCo, as well as adversarial training techniques, CLAE contributed to the growing body of research combining adversarial learning with SSL, paving the way for more advanced techniques like adversarial data augmentation in SSL.

Different from CLAE which actively generates adversarial examples to "attack" the model during training, forcing it to learn features robust to those specific attacks, Robust Contrastive Learning RoCL [108] takes a more defensive approach by using robust optimization. Instead of directly minimizing the contrastive loss on augmented views of an image, RoCL considers a "worst-case" scenario within a small perturbation set around each augmented view. It seeks to minimize the maximum contrastive loss that could be incurred by any perturbation within this set, effectively "defending" against a wider variety of possible variations, making it more robust. So, while both methods strive for robustness, CLAE is like training with specific attack simulations, while RoCL is like building a generally strong defence against any potential threat.

Self-Supervised Adversarial Training (SAT) [109] is a discriminative SSL method designed to enhance adversarial robustness. Unlike contrastive methods, SAT doesn't rely on contrastive loss or positive/negative pairs. Instead, it adapts traditional adversarial training to self-supervised settings, often using models like AMDIM as initial models. SAT generates adversarial examples via iterative gradient-based attacks, like Projected Gradient Descent (PGD), perturbing inputs within a defined bound. Specifically, it leverages pretext tasks (e.g., predicting image rotations or solving jigsaw puzzles) to create self-supervised pseudo-labels. Adversarial examples are then generated based on these self-supervised tasks, and the model is trained to maximize the mutual information between clean and perturbed samples, ensuring robustness against adversarial attacks.

Adversarial Contrastive Learning (AdCo) [110] is another discriminative SSL method that directly learns a set of negative adversaries to play against the self-trained representation network. It works by alternately updating two players: the representation network and negative adversaries, to obtain challenging negative examples against which positive queries are trained to discriminate. It updates negative adversaries towards a weighted combination of positive queries by maximizing the adversarial contrastive loss, allowing them to closely track representation changes over time. AdCo was influenced by existing contrastive learning methods like MoCo-v2, but improved upon them by more efficiently updating negative examples.

Cooperative-adversarial Contrastive (CaCo) [111] extends AdCo by jointly learning positive and negative representations end-to-end. While CaCo, likeAdCo, adversarially optimizes negative samples to maximize the contrastive loss, it cooperatively optimizes positive samples to minimize this loss, encouraging alignment with the query anchor and making them more challenging for the encoder. This dynamic interplay

ensures that both positive and negative samples adapt continuously to the evolving representations of the query anchors across mini-batches.

Decoupled Adversarial Contrastive Learning (DeACL) [112] employs a two-stage framework to improve adversarial robustness in SSL without using labels. In the first stage, DeACL performs standard SSL to obtain a non-robust encoder. In the second stage, this pretrained encoder acts as a teacher model, generating pseudo-targets to guide supervised adversarial training on a student model. This approach decouples the complex problem of integrating adversarial training into SSL into two simpler sub-problems: non-robust SSL and pseudo-supervised adversarial training.

Finally, DiRA (Discriminative, Restorative, and Adversarial Learning)[113] is an SSL framework that unifies three complementary components (i.e., discriminative, restorative, and adversarial learning), to collaboratively enhance representation learning. The discriminative component focuses on learning high-level semantic representations by maximizing agreement between similar instances in the latent space. The restorative component enforces the model to reconstruct fine-grained details of input images, ensuring that localized information is preserved. The adversarial component strengthens feature learning by introducing adversarial perturbations during restoration, encouraging the model to capture robust and generalizable representations. DiRA combines these components into a unified training framework with a joint loss function, enabling it to assemble complementary visual information from unlabeled data. DiRA was influenced by earlier SSL methods like MoCo-v2, SimSiam and Barlow Twins for its discriminative learning principles, as well as generative approaches for its restorative component. By combining these ideas, it bridges the gap between discriminative and generative SSL methods.

3.2.7 Methods Inspired from Diverse Fields

This subsection explores the cross-pollination of ideas from diverse fields, incorporating principles from areas outside traditional contrastive learning. By integrating concepts like max-margin classification or manifold learning, these methods aim to expand the capabilities of SSL.

Max-Margin Contrastive Learning (MMCL) [114] enhances contrastive learning by applying principles from Support Vector Machines (SVMs) to select high-quality negative samples and maximize the decision margin. MMCL computes a discriminative hyperplane for each positive example using an SVM objective, identifying the most informative negatives as support vectors. These "hard" negatives are then used to minimize their similarity with positive examples while maximizing positive pair similarity. To improve computational efficiency, MMCL simplifies the SVM objective using techniques like projected gradient descent or truncated least squares solvers, enabling end-to-end training. In contrast to methods like MoCo, SimCLR and DCL, MMCL requires neither large batch sizes nor momentum encoding, reduces training time, exhibits less sensitivity to hyperparameters, and improves generalization to downstream tasks like transfer learning and few-shot recognition.

Maximum Manifold Capacity Representations (MMCR) [115] takes a unique approach, drawing inspiration from "manifold capacity". Instead of contrasting image

views, MMCR aims to maximize the number of linearly separable categories within the learned representation space. Unlike MMCL, which uses SVM-based max-margin principles, MMCR optimizes the geometry of the entire representation space by aligning and uniformly distributing embeddings. In simpler terms, MMCR tries to arrange objects in a space so that lines can be easily drawn to separate them into different groups; the more groups that can be separated, the better the arrangement. MMCR aims to learn representations that make this separation easy. It achieves this by introducing a simplified manifold capacity objective, optimizing it to learn representations that compress object manifolds while preserving class separability. Specifically, MMCR uses a backbone and projector, generates multiple augmented views and processes them through both encoders to produce aggregated ℓ_2-normalized embeddings, calculates batch centroids, and optimizes the nuclear norm of the resulting centroid matrix. This approach enhances linear separability, allowing MMCR to match or surpass state-of-the-art SSL methods on benchmarks like linear evaluation.

Style Augmentations for Self-Supervised Learning (SASSL) [116] addresses the limitations of conventional data augmentation methods in preserving semantic information during self-supervised pretraining. Unlike traditional augmentation techniques that may distort the inherent structure of natural images, SASSL leverages Neural Style Transfer to decouple semantic and stylistic attributes in images. By applying transformations exclusively to the style component while preserving content, SASSL generates diverse samples that better retain semantic information. The method introduces a controlled stylization process using feature blending and image interpolation factors to ensure semantic consistency. SASSL offers two strategies for style references: external stylization using pre-computed representations from alternate datasets, and in-batch stylization utilizing other images from the same minibatch. When integrated into established self-supervised methods like MoCo, SimCLR, and BYOL, SASSL demonstrates significant improvements in ImageNet classification accuracy and transfer learning performance without increasing pretraining computational costs.

Table 11, provides a summary of the various contrastive learning methods discussed above, highlighting essential attributes that define and differentiate each approach. The table is structured to showcase the critical aspects of these methods, including the type of loss objective employed, the network architecture (whether siamese or single), specifics on how positive examples are defined ("views"), and the strategy for sampling negatives. Since the introduction of MoCo, features such as momentum encoders [16] and memory structures [59] have become integral to the design of these methods.

3.2.8 Dense Contrastive Methods

Dense contrastive methods in SSL address the limitations of traditional approaches by optimizing contrastive loss at the local feature or pixel level, making them ideal for dense prediction tasks like segmentation and object detection. Unlike methods that collapse spatial information into global image representations, dense methods employ dense projection heads to maintain spatial structure and compute the contrastive loss between corresponding local features across augmented views. Instance discrimination,

for example, uses global image-level feature representations obtained by average pooling the final convolutional feature map. This approach lacks spatial consistency in the convolutional features, which is crucial for tasks like object detection using architectures such as Faster R-CNN [117]. While image classification favors translation and scale invariance, reducing objects of various scales and locations to a single category, object detection requires translation and scale equivariance. The following section highlights the commonly employed SSL methods emphasizing dense contrastive learning. By retaining spatial details, dense contrastive methods enable better transferability to downstream applications requiring detailed spatial understanding, bridging the gap between self-supervised pre-training and tasks demanding pixel-level accuracy.

Dense Contrastive Learning (DenseCL) [118] extends the MoCo-v2 framework for dense prediction tasks by optimizing a pairwise contrastive (dis)similarity loss at the pixel level between two views of input images. DenseCL employs a specialized dense projection head to generate feature maps, enhancing the model's ability to process spatially detailed information. Adapting the InfoNCE loss to each dense local feature independently creates a more comprehensive learning objective, combining the traditional InfoNCE loss with a novel dense loss. Demonstrating the benefits of dense supervision, DenseCL is considered a key early contribution to dense contrastive learning.

View-Agnostic Dense Representation (VADeR) [119] is another early dense method that learns pixelwise representations by forcing local features to remain constant over different viewing conditions through pixel-level contrastive learning. It uses an encoder-decoder architecture and computes similarity on pixel-level representations, leveraging known pixel correspondences derived from the view generation process to match local features. Specifically, VADeR uses the InfoNCE loss, with temperature-calibrated cosine similarity at the pixel level.

Pixel-level consistency methods (PixContrast and PixPro) [120] employ a momentum encoder to distinguish each pixel within an image, treating each pixel as a distinct class. PixContrast identifies positive and negative pixel pairs based on their relative spatial locations, using a contrastive approach to emphasize spatial sensitivity in feature learning. Specifically, it considers pixels within a certain spatial proximity as positive pairs and pixels further apart as negative pairs. PixPro builds upon this by smoothing pixel features based on the similarity of neighboring pixels. A Pixel Propagation Module (PPM) aggregates information from surrounding pixels, refining each pixel's feature representation. Instead of explicit positive/negative pairs, PixPro emphasizes consistency between original and propagated (smoothed) pixel features, effectively capturing local spatial relationships without requiring explicit negative samples. This pixel-to-propagation consistency forms the basis of its learning objective.

Contrastive Detection (DetCon) [121] is built on top of BYOL and SimCLR, leading to DetConB and DetConS versions. DetCon introduces a novel self-supervised objective, tasking representations with identifying object-level features across augmentations. Using an unsupervised segmentation algorithm, DetCon generates masks for each view to identify approximate object regions in images. Mask-pooled representations, calculated via the dot product of masks and features, focus on these regions of interest identified through segmentation. Subsequent projection/prediction networks

generate latent features upon which the InfoNCE loss is applied. DetConS's key innovation is the incorporation of object-centric priors into SSL, making it well-suited for downstream detection tasks.

Region Similarity Representation Learning (ReSim) [122] learns both regional representations for localization and semantic image-level representations through self-supervision. It operates by sliding a fixed-size window across overlapping areas between two image views, aligning these areas with corresponding convolutional feature map regions and maximizing feature similarity. Specifically, ReSim uses the Region Proposal Network (RPN) from Faster R-CNN [117] to extract features from overlapping regions of the two views, sliding windows over the C3 or C4 feature maps (representing progressively higher-level abstractions, with C3 capturing localized details and C4 capturing global semantic information). ReSim's objective is to make common overlapping regions similar and non-common regions dissimilar by optimizing a weighted sum of local and global (InfoNCE) losses. ReSim's key contribution is learning spatially and semantically consistent feature representations throughout the convolutional feature maps, enabling downstream tasks to leverage these representations for localization.

Instance Localization (InsLoc) [123] extends the MoCo-v2 framework specifically for object detection pretraining. It creates composite images by pasting image instances at various locations and scales onto background images. Region of Interest Align (RoI Align) [124] is utilized to extract foreground features from the backbone feature maps. These features are then used with the InfoNCE loss to learn representations by treating different augmentations of the same object as positive pairs and features from different objects as negative pairs. Its key innovation is integrating bounding box information into pretraining, promoting better task and architecture alignment for transfer learning to detection. A subsequent case study [125] further explored InsLoc, proposing strategies to augment views and enhance robustness in appearance-shifted and context-shifted scenarios.

VICReg Large (VICRegL) [126] extends the original VICReg (Variance-Invariance-Covariance Regularization) SSL method, which aims to prevent representation collapse by applying three regularization terms: variance, invariance, and covariance (discussed further in Section 3.6). VICRegL builds upon this by introducing a local loss computed on patches extracted from full-resolution feature maps, in addition to the global loss computed on pooled features. The pipeline of VICRegL involves feeding two augmented views of an image through an encoder, producing both pooled and unpooled representations. These representations are then passed through an expander to generate embeddings. The method applies the VICReg loss to the global embeddings and introduces a feature vector matching process for local embeddings. This approach allows VICRegL to learn both global and local features simultaneously, improving performance on dense prediction tasks like segmentation while maintaining strong performance on classification tasks.

Copy-Paste Contrastive Pretraining (CPCP or CP2) [127], builds upon MoCo-v2 but extends it with a pixel-wise contrastive loss and a copy-paste mechanism. CP2 creates composite images by copying random foreground crops onto different backgrounds, training a segmentation model with two objectives: distinguishing foreground from background pixels and identifying composed images sharing the same

foreground. Using InfoNCE loss for both instance-level (applied to average foreground features) and pixel-level (applied to feature maps) contrastive learning, CP2 introduces a Quick Tuning protocol—initializing with pretrained weights and fine-tuning the entire segmentation model—to quickly adapt classification models to segmentation. While sharing copy-paste techniques with InsLoc, CP2 differs by focusing on dense prediction and using a segmentation head during pretraining, specifically targeting semantic segmentation (rather than object detection pretraining) with a dense contrastive loss for enhanced pixel-level representation learning.

Selective Object COntrastive learning (SoCo) [128] builds upon MoCo-v2 by introducing object-level representations to SSL, specifically designed for object detection tasks. SoCo utilizes off-the-shelf selective search to generate object proposals, treating each proposal as an independent instance rather than considering the whole image. This allows for a scale-aware assignment strategy where object proposals are assigned to different pyramid levels according to their size, encouraging scale-invariant representations. Unlike image-level methods, SoCo performs pretraining over all network modules used in detectors (backbone, FPN, and R-CNN head), enabling well-initialized weights for all detector layers. Through multiple augmented views with different scales and locations of the same object, SoCo learns object-level translation and scale invariance, achieving state-of-the-art transfer performance on COCO detection with significant improvements over supervised pretraining baselines.

DetCo (DetCo) [129] builds upon contrastive learning frameworks like MoCo-v2 to create a method that performs well on both instance-level detection tasks and image classification simultaneously. Unlike concurrent works that focused on detection but sacrificed classification performance, DetCo achieves a better trade-off through two key innovations: (1) multi-level supervision on features from different backbone stages, ensuring discriminative representations in each level of the feature pyramid, and (2) contrastive learning between global images and local patches using separate MLP heads. The global-local contrastive loss combines three components: global-to-global, local-to-local, and global-to-local, enabling consistent feature learning across scales. DetCo significantly outperforms methods like DenseCL and InsLoc on ImageNet classification while maintaining competitive performance on detection and segmentation tasks, demonstrating how detection and classification can mutually improve through proper architecture design.

Instance Consistency (InsCon) [130] builds upon the framework of MoCo-v2 to provide enhanced feature representations for downstream prediction tasks like object detection and instance segmentation. InsCon introduces three parallel learning heads to capture different levels of instance information: (1) the original single-instance head from MoCo-v2 for basic instance discrimination, (2) a multi-instance head that processes a composite image created from multiple augmented views arranged in a grid pattern, enabling the model to handle complex multi-instance recognition scenarios, and (3) a cell-instance head that focuses on fine-grained features by establishing strict cell-level correspondence between views with overlapping areas. The multi-instance head uses an adaptive average pooling layer to generate multiple feature representations, while the cell-instance head employs Precise ROI Pooling to extract matched cell features from overlapping areas between views. By combining these three learning

paradigms with a unified contrastive loss function, InsCon creates a complementary relationship between different granularities of feature representation, improving performance on object recognition, boundary localization, and content extraction in downstream tasks.

Unified Self-supervised Visual Pre-training (UniVIP) [131] is designed to learn versatile visual representations from both single-centric-object (e.g., ImageNet) and non-iconic (e.g., COCO [132]) datasets. UniVIP builds upon the BYOL framework, adopting its architecture and loss function as a foundation for scene-level consistency. However, UniVIP extends BYOL by incorporating instance-level learning and using an optimal transport algorithm to measure instance discrimination. The UniVIP pipeline involves three key steps: 1) Using the unsupervised Selective Search algorithm to generate candidate object instances, 2) Creating two scene views with overlapping regions containing instances to ensure global similarity, and 3) Applying representation learning at three levels: scene-scene similarity, scene-instance correlation, and instance-instance discrimination.

Table 12, summarizes the key architectural and operational attributes of the aforementioned dense contrastive methods.

3.3 Clustering Methods

Clustering-based discriminative SSL methods aim to learn representations by grouping similar samples together in the feature space. The main idea behind these methods is to leverage the inherent structure of the data to create pseudo-labels or assignments, which are then used to guide the learning process. Unlike contrastive methods that focus on instance-level discrimination, clustering-based approaches operate at a higher level by forming clusters of semantically similar instances. However, they share similarities with contrastive methods in that both aim to learn discriminative features without relying on manual labels. Clustering-based SSL methods typically use techniques like k-means or more sophisticated algorithms to assign samples to clusters, and then optimize the model to predict these cluster assignments or to align representations with learned prototypes. These methods appeared after contrastive SSL methods, building upon the success of earlier approaches and addressing some of their limitations, such as the need for large batch sizes or negative samples. Clustering-based methods like SwAV [64] have shown competitive performance with contrastive methods while offering unique advantages in terms of scalability and the ability to capture hierarchical semantic structures in the data. This section provides a description of the popular self-supervised methods centered around clustering.

Deep Cluster [67] is an early unsupervised learning method that jointly learns the parameters of a neural network and cluster assignments of the resulting features. It iteratively extracts features from images using a CNN, reduces their dimensionality, and clusters them using k-means. These cluster assignments then serve as pseudo-labels to update the network weights through backpropagation, alternating between feature clustering and network updates. While predating most contrastive SSL methods, Deep Cluster relies on the principle that good features should be clusterable, using cluster assignments as a form of self-supervision. Deeper Cluster [70] extends Deep Cluster

by incorporating a self-supervised pretext task—grouping features using pseudo-labels derived from angular rotations [46]—alongside the clustering objective. This combined approach leverages the benefits of both clustering-based and SSL, making Deeper Cluster [70] more effective for learning from large-scale, uncurated datasets compared to Deep Cluster.

Local Aggregation (LA) [133] aims to learn visual embeddings by optimizing a soft clustering structure in the feature space. Unlike Deep Cluster, which uses global clustering for pseudo-labels, LA identifies neighbors separately for each example, allowing for more flexible statistical structures. Specifically, LA iteratively extracts features using a CNN, identifies background and close neighbors for each sample, and optimizes a local aggregation metric that encourages similar samples to cluster (using k-means) together while separating dissimilar ones. LA employs a non-parametric softmax operation to compute sample recognition probabilities, which are then used in the objective function. Compared to Deep Cluster, LA does not require an additional readout layer or frequent recomputation of cluster labels, resulting in improved training efficiency.

Online Deep Clustering (ODC) [134] seeks to overcome the instability of alternating clustering and network updates in Deep Cluster by performing these processes simultaneously within each network update iteration. The training process consists of forward and backward propagations, label re-assignment, and centroid updates in each iteration. ODC maintains two dynamic memory modules: a sample memory to store sample labels and features, and a centroid memory for centroid evolution. This approach allows for continuous label updates rather than periodic global clustering, enabling more stable representation learning. Unlike Deep Cluster, ODC, similar to LA, does not require an additional readout layer or frequent recomputation of cluster labels, improving training efficiency. ODC also introduces loss re-weighting to prevent training collapse and a strategy for handling small clusters to prevent drifting into a few large clusters.

Self Labeling (SeLa) [135] also seeks to overcome Deep Cluster's limitation of alternating between clustering and network updates by optimizing a single objective function that combines both tasks. The process involves extracting image features using a CNN, applying a fast version [136] of the Sinkhorn-Knopp (SK) clustering algorithm to assign pseudo-labels, and then training the network to predict these labels using cross-entropy loss. SeLa introduces an equipartition constraint to prevent degenerate solutions, formulating the problem as an optimal transport task. The method can employ multiple projection heads for different clustering tasks, sharing the feature extractor parameters.

Constraint K-means (CoKe) [137] introduces online constrained k-means for clustering-based pretext tasks, effectively separating clustering from discrimination to optimize single-view instances. In the clustering phase, CoKe uses an online constrained k-means algorithm that only constrains the minimum size of each cluster (to avoid cluster collapse), enabling more flexible data structure capture compared to balanced clustering methods like SeLa. This online assignment method is theoretically guaranteed to approach the global optimum. In the discrimination phase, CoKe uses a standard normalized softmax loss with labels and centers recorded from the previous

epoch to learn representations. Unlike Deep Cluster, which requires offline batch-mode clustering, CoKe's online approach allows for more efficient training.

Swapping Assignments between multiple Views of the same image (SwAV) [64] is a novel SSL method combining clustering and contrastive learning. It employs Siamese networks to generate multiple augmented views of an image (Figure 8a), extracts features using a CNN, and computes cluster assignments (codes) for these features using learnable prototypes. The SK algorithm is utilized to efficiently compute these online cluster assignments, treating it as an optimal transport problem with an entropic constraint. SwAV introduces a "swapped" prediction mechanism, predicting the code of one view from the representation of another view of the same image, enforcing consistency between cluster assignments without explicit pairwise feature comparisons. This approach incorporates ideas from contrastive learning but replaces instance-level comparisons with cluster assignment comparisons. SwAV also introduces a multi-crop augmentation strategy for improved efficiency and performance and, unlike some other clustering-based methods, can be trained with both large and small batches and scales well to large datasets. SwAV further re-implemented and improved the SeLa and Deep Cluster frameworks, resulting in SeLa-v2 and Deep Cluster-v2, respectively. These versions incorporate stronger data augmentation, an MLP projection head, a cosine learning rate schedule, a temperature parameter, a memory bank, and multi-clustering, as introduced in SwAV.

SElf-supERvised learning (SEER) [17, 18] is a large-scale SSL approach that trains RegNet-Y [138] architectures using the SwAV method on random, uncurated internet images. SEER extracts features, applies online clustering to group similar visual concepts, and optimizes a contrastive loss between different views of the same image. Leveraging billions of random internet images without supervision, SEER scales SwAV's online clustering and contrastive learning principles to much larger models and datasets, efficiently handling extreme scale and diversity. Incorporating techniques like mixed precision training, gradient checkpointing, and sharded optimization, SEER distinguishes itself from other cluster-based methods through its ability to train at such scale.

Semantic Clustering by Adopting Nearest neighbors (SCAN) [139] introduces a novel two-step approach for unsupervised image classification. First, SCAN learns feature representations through a pretext task and mines nearest neighbors based on feature similarity, leveraging these neighbors as a semantic prior—unlike methods using k-means clustering after feature learning—and employing a self-labeling step to mitigate noise in nearest neighbor selection. Second, SCAN integrates this prior into a learnable approach, classifying each image and its mined neighbors together using a loss function that maximizes their dot product while encouraging consistent and discriminative predictions. SCAN's reliance on more meaningful features, rather than network architecture alone, distinguishes it from end-to-end approaches.

Prototypical Contrastive Learning (PCL) [63] (see Figure 8c) introduces proto-types as latent variables in an Expectation-Maximization (EM) framework, where high-confidence samples become class prototypes. Its two-step process involves finding the prototype distribution via clustering (E-step) and optimizing the network via contrastive learning using a novel ProtoNCE loss (M-step). This ProtoNCE loss, a

generalized InfoNCE loss, encourages representations closer to assigned prototypes while maintaining instance-level discrimination. Unlike SCAN, which uses nearest neighbors as a semantic prior, PCL directly learns prototypes through the EM framework. Building upon instance-wise contrastive learning, PCL incorporates semantic structures discovered through clustering, enabling it to learn both low-level features and higher-level semantics, improving performance, especially in low-resource transfer learning.

Synchronous Momentum Grouping (SMoG) [140] replaces instance-level contrastive learning with a group-level approach, effectively mimicking clustering-based methods within a contrastive learning framework. Using a Siamese network structure (with the first network comprising a backbone, projector, and predictor, and the second network comprising a backbone and projector, as shown in Figure 8b) with momentum-updated encoders, SMoG extracts features from two augmented views of an image. Instead of directly contrasting instance features, SMoG dynamically assigns instances to groups and performs contrastive learning at the group level. A momentum grouping scheme synchronously conducts feature grouping with representation learning, enabling gradient propagation through group features. This approach addresses the supervisory signal hysteresis issue faced by clustering-based methods and reduces false negatives common in instance contrastive methods.

Finally, Self-Classifier [141] is a novel, end-to-end self-supervised classification learning approach that simultaneously learns labels and representations in a single stage. It optimizes for same-class prediction of two augmented views of the same sample, using a shared network comprised of a backbone (e.g., CNN) and a classifier (e.g., projection MLP and linear classification head). To avoid degenerate solutions, Self-Classifier employs a mathematically motivated variant of the cross-entropy loss with a uniform prior asserted on the predicted labels. Unlike SCAN and other cluster-based SSL methods that often require pre-training, expectation-maximization, pseudo-labeling, or external clustering, Self-Classifier operates in a single stage without these additional steps. The pipeline is simpler and more scalable, processing two augmented views of an image through the shared network and minimizing their cross-entropy to promote same-class prediction (Figure 8d).

Clustering-based SSL methods offer several advantages over contrastive approaches. By forming clusters of similar instances, they can operate at a higher semantic level, potentially capturing more meaningful data structures. Often requiring smaller batch sizes and not relying on negative samples, they offer improved scalability and efficiency. Furthermore, they can better capture hierarchical semantic structures. However, clustering methods also have drawbacks. They can be more sensitive to data augmentation choices, as overly aggressive transformations can distort image information and hinder effective clustering. Additionally, being a more recent development than contrastive learning, they may not have benefited from the same level of extensive research and optimization. Some clustering approaches may also require complex design strategies, such as careful management of learnable prototypes. Despite these challenges, clustering-based methods have demonstrated competitive performance while offering unique advantages in scalability and semantic structure capture. Table 13 summarizes the clustering methods discussed above, highlighting their structural differences and
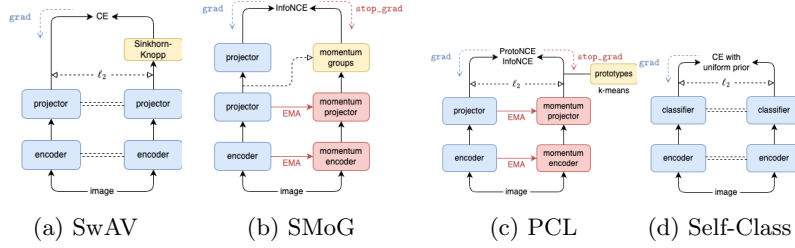
(a) SwAV   (b) SMoG   (c) PCL   (d) Self-Class

Fig. 8: Comparison of different clustering SSL architectures. (a) SwAV utilizes siamese networks to process two views of an image, aligning embeddings with trainable prototypes via the SK algorithm . (b) In SMoG , two networks learn synchronously with group assignments updated in real-time, using InfoNCE loss variants. (c) PCL uses prototypes as latent variables to guide the network towards effective group discrimination. (d) Self-Classifier replaces clustering in favor of direct label learning through a modified CE loss

specific features, including the presence of a projector or predictor, the use of momentum or memory structures, the incorporation of prototypes for guiding clustering, and the employment of the SK algorithm for distribution alignment. The loss objectives used by each method are also detailed.

3.4 Self-distillation Methods

Self-distillation methods in discriminative SSL improve representation learning by leveraging information from different parts of the same network. Unlike contrastive methods (which focus on instance-level discrimination) and clustering-based methods (which explicitly form clusters or prototypes), self-distillation methods prioritize internal consistency across different parts of the network or different augmentations of the same input. Learning without labeled data or negative samples, they often employ a teacher-student framework for knowledge transfer within the same model or between similar architectures. These methods typically use two encoders (or two views of the same encoder) processing differently augmented views of an image, with one encoder acting as a target for the other. To prevent encoder collapse, where a constant output is predicted for any input, techniques like updating one encoder's weights with a running average of the other's are commonly used. The following describes the most common self-distillation methods.

Bootstrap Your Own Latent (BYOL) [29], a seminal SSL method, uses two neural networks—an online network and a target network—to learn image representations without relying on negative pairs. The process involves passing two augmented views of an image through these networks to produce (ℓ_2)-normalized embeddings. The online network (comprising an encoder, projector, and predictor) is trained to predict the target network's representation of the same image under a different augmentation (see Figure 9a). The target network (sharing the same architecture as the online network,

but without the predictor) is updated as an EMA of the online network. The objective is to minimize the MSE between the online network's prediction and the target network's embedding. BYOL avoids representation collapse through its asymmetric architecture, predictor network, stop-gradient operation on the target network, and the EMA update mechanism, which together prevent the networks from converging to trivial solutions. While influenced by contrastive SSL, BYOL eliminates the need for negative pairs, instead focusing on predicting representations across different augmented views. This approach has proven more robust to image augmentation choices and achieves state-of-the-art performance on various benchmarks.

C-BYOL [86] advances BYOL by incorporating information compression using the Conditional Entropy Bottleneck (CEB), which allows for measuring and controlling the amount of information compression in the learned representation, leading to improved accuracy and robustness to domain shifts across various scenarios.

Self-DIstillation with NO labels (DINO) [142] is another seminal self-distillation method that uses a teacher-student architecture to learn visual representations without labels. The pipeline involves passing two augmented views of an image through student and teacher networks with identical architectures but different weights (see Figure 9b). The student network is trained to predict the teacher network's output using a CE loss, with the teacher's output centered and sharpened to avoid collapse. The teacher network is updated as an EMA of the student network. Similar to BYOL in its use of a momentum encoder and self-distillation, DINO differs by directly predicting the teacher's output (without a prediction head) and using a CE loss instead of MSE. DINO has shown particularly strong performance with ViTs, demonstrating "emergent properties" such as unsupervised segmentation through attention maps.

DINO-v2 [143], an improvement upon DINO, integrates features from DINO, iBOT [144], and SwAV, along with a new regularizer [145], to enhance training stability and speed. It introduces several innovations, such as masked input patches and untied head weights, aimed at improving learning efficiency and feature diversity.

Simple Siamese (SimSiam) [53] method learns representations by predicting one view of an image from another view of the same image. It does this passing two augmented views through a shared encoder network followed by a projection MLP (see Figure 9c). One branch then applies a prediction MLP, while the other uses a stop-gradient operation. The model is trained to maximize the cosine similarity between the predicted and stopped representations. SimSiam differs from BYOL by eliminating the need for a momentum encoder, and from DINO by not using a teacher-student framework or temperature-based softmax. The stop-gradient operation and predictor MLP, and the use of NCS loss, are crucial for preventing collapse and enabling effective learning without negative samples or momentum encoders. Fast Siamese (FastSiam) [146] advances SimSiam by introducing a multi-crop strategy and a novel loss function that combines intra-image and inter-image consistency, allowing for faster convergence and improved performance.

Finally, Online Bag-of-Visual-Words (OBoW) [147] uses a teacher-student architecture to learn visual representations without labels by processing two augmented views of an image through the teacher and student networks. The teacher network generates a Bag-of-Visual-Words (BoW) representation (a technique representing images as

histograms of quantized local features, analogous to word histograms in text analysis) using an online-updated vocabulary of visual words. The student network is trained to predict this BoW representation from a different augmented view. OBoW performs online training of both networks and continuously updates the visual-word vocabulary, enabling fully online BoW-guided unsupervised learning. Unlike BYOL, SimSiam, and DINO, which focus on making global image embeddings invariant to augmentations, OBoW uses high-dimensional BoW vectors as targets, capturing multiple local visual concepts and aiming to promote contextual reasoning.

Table 14 summarizes the self-distillation methods discussed, illustrating their use of projectors, predictors, and momentum encoders. A key feature common to all these methods is the `stop-grad` operation, crucial for preventing network collapse into trivial solutions, and the lack of memory use. The "networks" column specifies the architectural setup, noting the different roles (e.g., "online/target" or "student/teacher") used in these Siamese network variations. The table also details the loss functions and symmetry of each method, providing insights into their specific optimization mechanisms.

In summary, self-distillation SSL methods offer several distinct advantages. While all SSL methods are sensitive to augmentations, self-distillation, by focusing on internal consistency, can be argued to be less reliant on carefully engineered augmentations than contrastive or clustering methods. This focus on internal consistency also simplifies training and reduces computational costs by avoiding the complexities of momentum encoders, negative samples, large memory queues, and complex clustering, making some self-distillation methods, like SimSiam, surprisingly simple to train. Furthermore, this focus can encourage more robust and meaningful representations, sometimes leading to remarkable emergent properties like unsupervised object segmentation, as seen in DINO. Finally, self-distillation methods are often more computationally efficient than traditional contrastive approaches, not requiring large batch sizes or memory banks, and have shown promise for improving performance on smaller datasets and models. While they technically do not require large batch sizes, they typically employ them in practice to achieve optimal performance. However, self-distillation methods also have disadvantages. They are susceptible to collapse, even with mitigation strategies, and can be sensitive to architectural and hyperparameter choices, often requiring careful tuning. Compared to contrastive learning, self-distillation is less understood theoretically, hindering the development of new methods and improvements to existing ones. Finally, there is a potential for redundancy in learned features, as different parts of the network might learn similar representations, which can limit the overall information captured. Like other SSL methods, self-distillation faces open challenges, including improving performance on smaller models, developing efficient training for large-scale datasets, and enhancing transferability to diverse downstream tasks.

3.5 Knowledge Distillation Methods

Knowledge distillation methods in SSL leverage a pre-trained "teacher" model to guide the learning of a "student" model, even when labeled data is scarce or unavailable [148]. Unlike self-distillation, which uses different parts of the same network (or similar
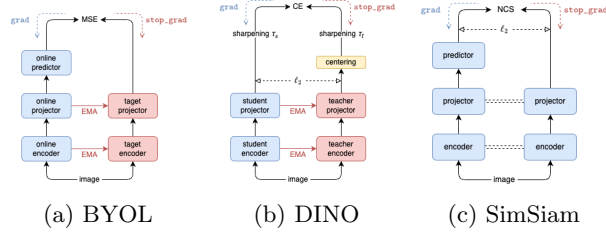
(a) BYOL	(b) DINO	(c) SimSiam

Fig. 9: Comparison of different self-distillation SSL architectures. (a) BYOL employs a dual network architecture where the target network's parameters are updated using an EMA of the online network's parameters. (b) DINO uses a student-teacher setup where the student learns to replicate the teacher's output, with the teacher's parameters being a EMA of the student's. (c) SimSiam operates two identical networks processing different views of the same image, using a stop-gradient operation to prevent output collapse.

architectures for both teacher and student) to provide the learning signal, knowledge distillation employs a separate, typically larger or more powerful, teacher network. This teacher network, pre-trained using any available (even limited) labeled data or through a self-supervised task, provides "soft" targets (e.g., probability distributions or feature representations) to the student network. The student network, which can be smaller or have a different architecture, learns to mimic the teacher's behavior, effectively distilling the learned knowledge. This differs significantly from self-distillation, which relies on internal consistency within a single network. The main advantages of knowledge distillation in SSL include improving the performance of smaller models, enabling SSL in resource-constrained environments, and transferring rich representations from large models to more efficient architectures. These methods address the challenge of SSL's ineffectiveness with smaller models, which limits its applicability in edge devices and other computationally constrained scenarios. The following provides an overview of typical approaches used in self-supervised knowledge distillation.

Contrastive Representation Distillation (CRD) [149] offers a novel approach to knowledge distillation by leveraging contrastive learning objectives to transfer representational knowledge between networks. Unlike traditional knowledge distillation methods that minimize KL divergence between output probabilities, CRD focuses on maximizing MI between teacher and student representations. The method introduces a framework that treats knowledge transfer as a binary classification problem. CRD uses a contrastive loss that pushes representations of the same input closer together while pushing representations of different inputs apart.

Relational Representation Distillation (RRD) [150] approaches self-supervised knowledge distillation by addressing the limitations of existing contrastive learning methods. Unlike traditional approaches that minimize KL divergence or use rigid instance-discrimination techniques, RRD focuses on preserving the relative relationships between instances in the feature space. The method introduces separate temperature parameters for teacher and student distributions. By allowing sharper

student outputs, RRD can precisely learn primary relationships while maintaining secondary similarities between instances. This approach goes beyond direct embedding transfer, capturing the broader structural relationships in the representation space, similar to ReSSL. This approach ensures that the self-supervised objective captures not just individual instance characteristics, but the broader structural relationships between instances. Theoretically connected to both InfoNCE loss and KL divergence, RRD aims to transfer knowledge more comprehensively.

Discriminative and Consistent Representation Distillation (DCD) [151] combines contrastive learning with consistency regularization to create a unique self-supervised knowledge distillation approach. The method introduces learnable temperature and bias parameters that adapt during training, allowing for more flexible self-supervised feature transfer. By balancing discriminative power with structural consistency, DCD uses self-supervised objectives to ensure that the student model not only mimics the teacher's individual instance representations but also captures the broader distributional characteristics of the teacher's learned features.

Multi-Mode Online Knowledge Distillation (MOKD) [152] leverages SSL by enabling two models to learn collaboratively through both self-distillation and cross-distillation modes. Unlike traditional knowledge distillation, MOKD uses self-supervised contrastive learning to help heterogeneous models (like CNNs and ViTs) extract complementary features. The method's cross-distillation mechanism allows models to transfer knowledge through a self-supervised objective, focusing on enhancing semantic feature alignment across different model architectures.

Self-Supervised Knowledge Distillation (SSKD) [153] directly incorporates self-supervision as an auxiliary task for knowledge transfer. By using contrastive prediction as a supplementary learning objective, SSKD extracts richer knowledge from the teacher network. The method explores various self-supervised tasks (contrastive learning [28], exemplar [154], jigsaw [10], and rotation prediction [46]) to create a more comprehensive knowledge transfer approach. Its selective transfer strategy ensures that only meaningful self-supervised signals are used, making it particularly effective in extracting structured knowledge from self-supervised representations.

Distill-on-the-Go (DoGo) [155] employs a unique approach to self-supervised knowledge distillation through mutual learning. It uses contrastive representation learning as its core self-supervised objective, where two models generate embeddings from augmented views of the same sample. The method's innovative contribution lies in aligning the softmax probabilities of similarity scores between peer models, effectively using self-supervised contrastive learning as a mechanism for knowledge transfer. Unlike other methods, DoGo focuses on collaborative learning where models simultaneously improve each other's representations.

SElf-SupErvised Distillation (SEED) [156] is an early method to introduce knowledge distillation to SSL, addressing the challenge of poor performance in smaller models trained directly with contrastive SSL methods. It is designed to improve the performance of smaller models by distilling knowledge from a larger pre-trained teacher model to a smaller student model without using labeled data (see Figure 10a). SEED first trains a large teacher model using a self-supervised approach like MoCo-v2 or SimCLR. Then, during distillation phase, SEED maintains a queue of data samples

and computes softmax-transformed similarity scores between an input instance and the queued samples using both teacher and student networks. The student is trained to mimic the teacher's similarity score distribution, minimizing the cross-entropy between these distributions.

Distilled Contrastive Learning (DisCo) [157] also trains a large teacher model using a self-supervised approach and then distills knowledge to a smaller student model. Unlike SEED, which mimics similarity score distributions, it directly distills the teacher's final embeddings to the student. DisCo has two variants: a Siamese version where teacher and student process the same input (Figure 10c), and a momentum version using a sample queue similar to MoCo (Figure 10d). DisCo incorporates a "mean student" encoder, used alongside the student and teacher encoders in computing the contrastive and consistency losses, which helps stabilize training and improve distilled knowledge quality. Specifically, DisCo processes two augmented views of an image through all three networks, calculating a contrastive loss between student and mean student embeddings from different views, and a distillation loss as the sum of MSE between student and teacher embeddings for both views, encouraging the student to mimic the teacher's representations. DisCo's main novelty is directly transferring the teacher's latent manifold, enabling more effective knowledge transfer, especially for lightweight models, and addressing the "distilling bottleneck" by enlarging the embedding dimension. Influenced by contrastive SSL, DisCo incorporates ideas from MoCo-v2 and SwAV, adapting them for distillation (e.g., teacher pre-training).

Bag of InstaNces aGgregatiOn (BINGO) [158] also uses student and teacher networks, each with a backbone and projector (see Figure 10b). However, unlike SEED, which distills similarity score distributions, and DisCo, which distills final embeddings, BINGO constructs bags of similar instances using teacher embeddings (pretrained with methods like MoCo-v2 or SwAV). Each bag contains an anchor instance and other highly similar instances. It then applies a bag-aggregation distillation loss, comprising intra-sample distillation (pushing together embeddings of two augmentations of the same instance) and inter-sample distillation (pushing embeddings of all bag instances toward the anchor). BINGO's main novelty is transferring relationships among similar samples, rather than simply mimicking teacher outputs or embeddings.

Table 15 compares the aforementioned methods, highlighting their diverse loss functions (e.g., softmax with CE, combinations of InfoNCE and MSE) for minimizing feature representation discrepancies between student and teacher. These methods are heavily influenced by Hinton et al. work [148], which, while focused on supervised learning, introduced the core concepts of knowledge distillation and demonstrated its effectiveness in transferring knowledge from larger teacher models to smaller student models, laying the foundation for knowledge distillation in SSL. Knowledge distillation methods offer a number of advantages over self-distillation and contrastive methods. Compared to self-distillation, they can transfer knowledge from larger, more powerful teacher models, potentially improving performance on resource-constrained devices. Unlike contrastive methods, they do not require large batch sizes or negative samples, increasing efficiency and flexibility. Moreover, they often perform better on smaller datasets, especially with noisy or limited labels. However, knowledge-distillation methods typically require a two-stage process: pre-training the teacher and
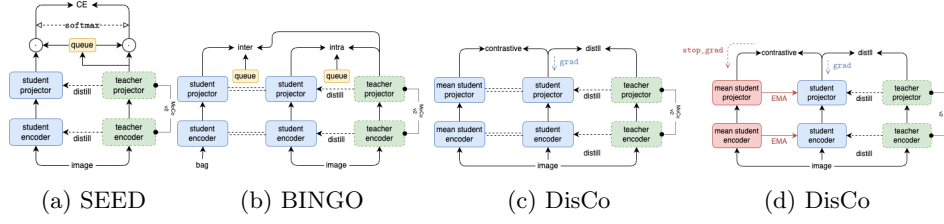
Fig. 10: Comparison of different knowledge-distillation SSL architectures. (a) SEED utilizes a teacher-student architecture where the student network learns from the teacher by minimizing the cross-entropy loss between their softmax-transformed embeddings. (b) BINGO aggregates features from an anchor image and similar instances into a bag of features, then processes them through both student and teacher networks, applying combined intra- and inter-contrastive losses. (c, d) DisCo employs a triplet of networks — student, mean student, and teacher — where the student learns from both its average self and the pretrained teacher through a combination of contrastive and distillation losses. DisCo can either use Siamese network that share parameters (c) or use momentum on the mean student (d).

then distilling knowledge to the student, which can be more time-consuming than end-to-end approaches. While generally less computationally expensive during inference, they may require more resources during training due to this two-stage process. Common challenges include designing effective distillation tasks, handling domain shifts, and balancing model compression and performance. Furthermore, the choice of teacher model significantly impacts student performance, and achieving consistent performance across different architectures remains a challenge.

3.6 Feature Decorrelation Methods

The self-supervised feature decorrelation family proposes novel losses to address the representation learning paradigm with self-supervision [7]. As a subset of it, the canonical correlation analysis family originates with the Canonical Correlation Analysis (CCA) [159]. The high-level goal of CCA is to infer the relationship between two variables by analyzing their cross-covariance matrices [68]. Feature decorrelation methods focus on creating representations that are diverse and independent, aiming to enhance the robustness and generalizability of learned features. By reducing redundancy among features, these approaches aim to improve the robustness and generalization of models. The following is a description of the most popular feature decorrelation methods.

Barlow Twins [68] (named after the neuroscientist Horace Barlow) employs a novel approach to SSL by leveraging the redundancy reduction principle, originally proposed in neuroscience. This method utilizes two identical Siamese networks, each consisting of a backbone and a projector, to process two different views of the same image generated through stochastic augmentation (see Figure 11a. In this setup, each image view is transformed into embeddings by the encoder network, which are then mean-centered

along the batch dimension to ensure an average output of zero over the batch, effectively normalizing the embeddings (i.e., ℓ_1-normalized). Barlow Twins to minimize a unique loss function based on the cross-correlation matrix between the embeddings of the two views. This loss function aims to ensure embeddings remain consistent across augmentations while minimizing redundancy among their components, effectively promoting invariance and feature decorrelation. Barlow Twins stand out among other methods as they do not demand large batch sizes SimCLR, nor rely on asymmetric components like prediction networks BYOL, SimSiam, momentum encoders MoCo, MoCo-v2, BYOL, or non-differentiable operations or stop-gradients SimSiam. It benefits from employing high-dimensional embeddings.

Variance-Invariance-Covariance Regularization (VICReg) [160] uses two regularization terms applied to embeddings separately. It employs two networks for learning that are not required to share weights or the same architecture, as shown in Figure 11b. Each network comprises a backbone and an expander serving as the projection head. Given an image, VICReg generates two views using stochastic augmentations. These views are processed by the backbone and expander of each network to produce embeddings. To ensure robust and informative feature learning, VICReg incorporates two regularization terms: one that maintains the variance of each embedding dimension above a specified threshold and another that decorrelates each pair of embedding variables using the off-diagonal elements of the covariance matrix, similar to Barlow Twins. Additionally, VICReg incorporates the MSE between the embeddings of the different views to prevent the model from collapsing into trivial solutions where the embeddings become constant or non-informative across different inputs. This dual regularization approach ensures that the variance of each embedding dimension is maintained while decorrelating variables, effectively preventing encoder collapse and ensuring that the learned representations remain meaningful and distinct.

Whitening Mean Squared Error (W-MSE) [161] is a loss function that is based on the whitening of the latent space features. The framework utilizes a network that comprises of a backbone and a projector, as depicted in Figure 11c. It derives multiple positive pairs from a single image, deeming negatives unnecessary. These views are processed by the network to produce the embeddings. The embeddings are mapped onto a spherical distribution using a whitening transform and then undergo ℓ_2-normalization. The normalized whitened representations are used to compute the MSE implemented as NCS over all positive pairs. W-MSE conceptually aligns with Barlow Twins. It diverges from approaches using momentum networks BYOL or stop-gradient networks SimSiam, promotes non-degenerate distributions through a whitening operation that normalizes representations to a spherical distribution with zero mean and identity covariance matrix. This method avoids the use of negative pairs by bringing positive pairs closer together, using batch slicing to manage the variability of the whitening matrix.

Subsequent developments like Mixed Barlow Twins, TLDR, and ARB build upon Barlow Twins' framework, incorporating distinctive elements to refine and stabilize the learning. Mixed Barlow Twins tackles the potential decrease in feature quality with higher embedding dimensions by enhancing sample interactions and adding a regularization term to prevent overfitting. TLDR and ARB expand on the primary concept

by incorporating nearest neighbors and the Nearest Orthonormal Basis (NOB) into their methodologies, respectively. More precisely, Mixed Barlow Twins [162] enhances the original Mixed Barlow framework by incorporating linearly interpolated samples to mitigate feature overfitting. Utilizing two identical Siamese networks, this approach processes two distinct views of the same image, generated via stochastic augmentation. A third, mixed view is also created by linearly blending these two augmented views. Mixed Barlow Twins retains the objective of Barlow Twins, supplementing it with a regularization component. This is achieved by aligning the cross-correlation matrix of the mixed view with that of each original view against the respective ground truth cross-correlation matrices. The overall loss for Mixed Barlow Twins combines the original Barlow Twins objective with this additional regularization term. Align Representations with Base (ARB) [163] builds on the Barlow Twins framework but uses the NOB [164] as an objective to maximize the MI between the immediate variable generated from one view and representations of the other view. Specifically, it minimizes the output matrix of one view and the nearest orthonormal basis matrix of the other view. Twin Learning for Dimensionality Reduction (TLDR) [165] builds on the Barlow Twins framework but employs the nearest neighbor along a view as a pair whose proximity is preserved for the purpose of dimensionality reduction.

Different from the above, Twin Class Distribution Estimation (TWIST) [166] utilizes Siamese networks, concluding with a softmax operation to yield twin class distribution probabilities for two augmented versions of the same image (see Figure 11d). It aims for consistency across these class distributions by maximizing MI between the inputs and class predictions. TWIST further sharpens the class distribution for each image to promote decisive class assignments and fosters diversity among predictions to avoid biased classifications towards a dominant class, achieving a balanced and accurate classification regime. Finally, Truncated Triplet [167] builds upon the BYOL framework, introducing a truncated triplet loss which extends the conventional triplet loss [58]. This new loss function aims to maximize the relative distance between positive pairs and negative pairs, addressing issues of under-clustering and over-clustering common in self-supervised settings.

Orthogonality Regularization (OR) [168] applies regularization across the encoder, targeting both convolutional and linear layers during pretraining. By enforcing orthogonality within weight matrices, OR prevents dimensional collapse—a phenomenon where a few large eigenvalues dominate the eigenspace—ensuring that filters remain uncorrelated and maintain a norm of 1. This regularization indirectly stabilizes hidden features and representations, promoting uniform eigenvalue distributions throughout the network. Leveraging two key regularizers, Soft Orthogonality (SO) and Spectral Restricted Isometry Property (SRIP), OR integrates seamlessly with existing SSL frameworks, enhancing performance.

In Table 16, we outline the key components of various feature decorrelation methods, highlighting their similarities and differences. Most methods listed employ a projector and utilize a Siamese architecture, reflecting a common framework for this category of methods. The loss functions employed by these methods are notably distinct and innovative, as they work directly on the features themselves to enhance

46

(a) Barlow Twins   (b) VICReg   (c) W-MSE   (d) TWIST
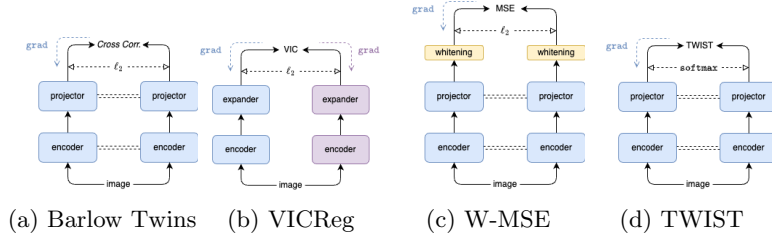
Fig. 11: Comparison of different feature decorrelation SSL architectures. (a) Barlow Twins utilizes two siamese networks aiming to minimize redundancy across the embeddings by aligning the cross-correlation matrix to the identity matrix. (b) VICReg employs a variance-invariance-covariance regularization strategy on the embeddings from two distinct views processed through separate networks. (c) W-MSE implements a whitening MSE approach to normalize embeddings into a spherical distribution. (d) TWIST utilizes twin siamese networks to generate class distribution probabilities for two augmented views of the same image, aiming for consistency across these distributions.

decorrelation. For instance, methods like Barlow Twins and TLDR use a cross-correlation matrix to minimize redundancy between features, while VICReg employs a combination of variance, invariance, and covariance regularization to achieve feature decorrelation. This table provides a clear view of how each method approaches the challenge of reducing feature redundancy to enhance model generalizability and robustness.

4 Benchmarking of Self-supervised Learning Methods

Self-supervised evaluation is crucial for assessing the quality and transferability of learned representations in computer vision tasks. It provides a standardized framework for comparing different SSL methods and their effectiveness in capturing meaningful features from unlabeled data. The main evaluation frameworks include linear evaluation, where a linear classifier is trained on frozen features to measure representation quality; semi-supervised fine-tuning, which assesses performance with limited labeled data; and transfer learning to downstream tasks such as object detection and segmentation. The importance of SSL evaluation lies in its ability to validate the generalization capabilities of self-supervised models, their data efficiency in low-label regimes, and their potential to reduce reliance on large-scale labeled datasets. As SSL continues to advance, robust evaluation protocols ensure that progress in the field translates to real-world improvements across various computer vision applications. This section details the standard protocols used to assess SSL-trained model performance, focusing on how these representations perform under various evaluation setups. We also compiled and summarized performance results from relevant papers. This section is organized as follows. First, we examine evaluation protocols on the source dataset in Section 4.1, covering both linear evaluation and semi-supervised approaches. Next,

we analyze transfer learning capabilities to other classification tasks in Section 4.2, followed by an assessment of transferability to more complex vision tasks such as detection and segmentation in Section 4.3. Throughout each subsection, we present both the methodological frameworks and comparative results across various self-supervised learning approaches.

4.1 Evaluation on the Source Dataset

This section describes the evaluation protocols employed to measure the effectiveness of the SSL method on the source dataset. This process involves training a model on unlabeled data using SSL techniques, then evaluating its performance on the same dataset using labeled samples. Common evaluation methods include linear probing, where a linear classifier is trained on frozen features, and fine-tuning, where the entire model is updated using limited labeled data. These evaluations provide insights into the model's ability to capture meaningful features from unlabeled data, its generalization capabilities, and its potential for downstream tasks.

4.1.1 Evaluation Protocols

Linear evaluation in SSL assesses representation quality by freezing a pretrained encoder (i.e., the backbone), training only a linear classifier on labeled data, and measuring classification accuracy on held-out test sets to determine feature separability and transferability. It was popularized by SSL frameworks like SimCLR, MoCo, and BYOL, and previous studies [169], although the concept of linear evaluation existed in earlier forms.

More specifically, linear evaluation includes the following steps: (1) a model (e.g., ResNet, ViT) is pretrained on unlabeled data using SSL objectives; (2) the pretrained encoder's weights are frozen to isolate representation quality; (3) a single linear layer (or MLP) is appended to the frozen encoder and trained exclusively on labeled data (e.g., ImageNet) using CE loss and Stochastic Gradient Descent (SGD) [170] with Nesterov momentum, typically for up to 100 epochs, without regularization - during this phase, only basic augmentations are applied for preprocessing: training images undergo resizing to 256×256, random resized cropping to 224×224, horizontal flipping, and normalization, while validation images undergo resizing to 256×256, center cropping to 224×224, and normalization, excluding heavy pretraining augmentations like color distortion; (4) the model's quality is evaluated by measuring classification accuracy on a held-out test set (e.g., ImageNet validation set), where higher accuracy indicates better feature separability and generalizability.

Linear evaluation assesses two critical properties of SSL representations: linear separability (whether features naturally cluster by semantic class) and transferability (whether representations generalize to downstream tasks without task-specific tuning), all while isolating feature quality by freezing the encoder. The protocol has gained widespread adoption due to its simplicity (requiring minimal computational resources), reproducibility (enabling direct comparison across SSL methods), and theoretical alignment with the principle that quality representations should permit simple linear decision boundaries. By focusing solely on the linear classifier's performance

48

atop frozen features, this method provides a controlled benchmark that distinguishes representation quality from fine-tuning capabilities, though it may not fully capture performance on complex non-linear tasks.

Semi-supervised evaluation assesses the quality of learned representations by combining SSL pretraining with limited labeled data during downstream task adaptation, following protocols from [171, 172]. The method emerged alongside SSL breakthroughs, driven by frameworks like SimCLR and BYOL and has become a cornerstone for measuring SSL's label efficiency and generalization capabilities. While linear evaluation (training a classifier on frozen features) was standardized earlier, semi-supervised evaluation gained traction as a more practical benchmark for real-world scenarios where labels are scarce.

Semi-supervised evaluation assesses the label efficiency and transferability of SSL representations through a three-stage pipeline: (1) a backbone model (e.g., ResNet, ViT) is pretrained on unlabeled data using SSL objectives; (2) semi-supervised fine-tuning integrates a small labeled subset (e.g., 1-10% of ImageNet) with the remaining unlabeled data, and the encoder weights are updated using these limited labels; (3) the fine-tuned model's performance is evaluated using classification accuracy on held-out test sets, such as the ImageNet validation set.

Fine-tuning and linear evaluation differ fundamentally in their methodological approach and evaluation focus. Fine-tuning updates all or most pre-trained encoder weights using smaller learning rates (e.g., 0.01-0.001) with layer-wise learning rate decay to prevent catastrophic forgetting of pretrained features, adapting the feature extraction for the specific downstream task, and evaluating the adaptability of representations by measuring how well the pre-trained features serve as initialization for task-specific learning compared to random initialization. Similar spatial augmentations are applied during both training and testing phases. Optimization of the CE loss continues under the same conditions as in the linear setup. In addition to exploring various learning rates, the duration of training epochs is also varied to determine the optimal settings that yield the best results on a local validation set (usually the epochs vary between 30 and 50). Conversely, linear evaluation freezes the entire pre-trained encoder backbone, training only a linear classifier layer with high learning rates to assess the linear separability of learned representations, essentially testing if the SSL objective has created a feature space where classes naturally cluster. Thus, linear evaluation focuses on the inherent quality of the learned representations, while fine-tuning assesses their adaptability to specific tasks.

4.1.2 Datasets

The most commonly used dataset utilized as the primary source for training and evaluating SSL models is ImageNet ILSVRC-2012 (also known as ImageNet-1K), a subset of the larger ImageNet-22K [19] database that has become a standard benchmark in computer vision. It contains 1,000 object categories organized according to the WordNet hierarchy, with a total of 1,281,167 training images, 50,000 validation images, and 100,000 test images. ImageNet-1K features RGB images of varying resolutions that were collected by scraping online image search engines using synonyms in multiple

Table 1: Source datasets commonly used for self-supervised pretext task training and evaluation.

Dataset	Classes	Images (train/validation)
ImageNet-1K [19]	1,000	1,281,167 / 50,000
ImageNet-100 [75]	100	130,000 / 5,000
ImageNet-22K [19]	21,841	14,197,122 / 50,000
Instagram-1B [175]	-	1,000,000,000 / -
iNaturalist18 [176]	8,142	437,513 / 24,426
CIFAR-10 [173]	10	50,000 / 10,000
CIFAR-100 [173]	100	50,000 / 10,000
Places205 [174]	205	2,500,000 / 20,000

languages. The images are quality-controlled and human-annotated, with each image labeled with exactly one class identifier. To facilitate a broader evaluation, datasets such as CIFAR-10, CIFAR-100 [173], Places205 [174], Instagram-1B [175] (private dataset), ImageNet-22K (a superset of ImageNet-1K) [19], ImageNet-100 (a subset of ImageNet-1K as proposed in [75]), and iNaturalist18 [176] are also employed. Table 1 details these datasets. The effectiveness of the representations learned through SSL is typically evaluated using linear evaluation and fine-tuning techniques, following the same protocol as [171, 172].

4.1.3 Comparative Results: Linear Evaluation on ImageNet-1K

We collected and compiled performance results of SSL methods in linear evaluation tasks using a ResNet-50 [49] encoder for fair comparison. Table 2 shows the top-1 and top-5 accuracies for various SSL methods under linear evaluation on ImageNet-1K, while Table 3 shows methods that employ multi-crop augmentation. We present methods employing multi-crop augmentation separately, as it is fair to do so due to the performance boost it offers, albeit at the cost of much longer training. Notably, ReLIC-v2 achieves the highest top-1 accuracy of 77.1% and a top-5 accuracy of 93.3%, demonstrating its strong performance in extracting useful representations. This result is attributed to the innovative approach of combining an explicit invariance loss with a contrastive objective over a varied set of appropriately constructed data views, as proposed by ReLIC. This method effectively avoids learning spurious correlations and obtains more informative representations, outperforming previous self-supervised approaches by significant margins. CLSA follows closely with a top-1 accuracy of 76.2%, benefiting from the strong augmentations it employs. AdCo and C-BYOL also perform well with top-1 accuracies of 75.7% and 75.6%, respectively. These results indicate that methods such as ReLIC-v2 and CLSA are particularly effective in learning high-quality representations that generalize well in a linear evaluation setup.

Table 2: Performance comparison of various SSL methods under linear evaluation protocol on ImageNet-1K using ResNet-50 backbone.

Method	Projector	Batch size	Epoch	Top-1	Top-5
Supervised	None	256	90	76.5	93.0
AdCo [110]	MLP	256	800	72.8	-
All4One [101]	MLP	1024	1000	66.6	87.5
ARB [163]	MLP	2048	100	68.2	88.9
Barlow Twins [68]	MLP	2048	1000	73.2	91.0
BYOL [29]	MLP	4096	1000	74.3	91.6
CaCo [111]	MLP	4096	800	75.3	-
C-BYOL [86]	MLP	4096	1000	75.6	92.7
CLSA [177]	MLP	256	800	72.2	-
CMC [75]	Linear	128	200	60.0	82.3
CoKe [137]	MLP	256	800	71.4	-
CPC-v2 [81]	MLP	512	200	63.8	85.3
C-SimCLR [86]	MLP	4096	1000	71.6	90.5
DeepCluster [67]	MLP	256	400	52.2	-
InfoMin [76]	MLP	256	800	73.0	91.1
InstDis [59]	Linear	256	200	54.0	-
InstDis++ [77]	Linear	256	800	59.0	-
Jigsaw [10]	None	256	90	45.7	-
LA [133]	Linear	256	200	60.2	-
Mixed Barlow Twins [162]	MLP	256	300	72.2	-
MMCL [114]	MLP	256	200	63.8	-
MoCHI [92]	MLP	256	800	68.7	-
MoCLR [99]	MLP	4096	1000	74.3	92.2
MoCo [16]	Linear	256	200	60.6	-
MoCo-v2 [88]	MLP	512	800	71.1	-
MoCo-v2 + DCL [60]	MLP	256	200	67.6	-
MoCo-v3 [89]	MLP	4096	800	73.8	-
MSF [100]	MLP	256	200	72.4	-
NNSiam [61]	MLP	256	4096	72.6	90.4
OBoW [147]	MLP	256	200	73.8	-
ODC [134]	MLP	256	400	55.7	-
PCL [63]	None	256	200	61.5	-
PCL-v2 [63]	Linear	256	200	67.6	-
PIRL [77]	Linear	1024	800	63.6	-
ReLIC [62]	MLP	4096	1000	70.3	89.5
ReLIC-v2 [105]	MLP	256	1000	**77.1**	93.3
ReSSL [97]	MLP	4096	200	69.9	-
RotNet [46]	None	256	256	48.9	-
SeLa [135]	Linear	256	400	61.5	84.0
Self-Classifier [141]	MLP	4096	800	74.1	-
SimCLR [28]	MLP	4096	1000	69.3	89.0
SimCLR + DCL [60]	MLP	256	200	65.8	-
SimCLR + DCLW [60]	MLP	256	200	66.9	-
SimCLR-v2 [82]	MLP	4096	800	71.7	90.4
SimSiam [53]	MLP	256	800	71.3	-
SMoG [140]	MLP	4096	800	<u>74.5</u>	91.9
SynCo [93]	MLP	256	800	70.7	-
TiCo [96]	MLP	4096	1000	73.4	91.6
TWIST [166]	MLP	4096	800	72.6	91.0
UniGrad [106]	MLP	4096	100	70.3	-
UniVIP [131]	MLP	4096	300	74.2	-
VICReg [160]	MLP	4096	1000	73.2	91.1
VICRegL [126]	MLP	256	300	71.2	-
W-MSE [161]	MLP	4096	400	72.5	-

Table 3: Performance comparison of SSL methods employing multi-crop augmentation under linear evaluation protocol on ImageNet-1K using ResNet-50 backbone.

Method	Projector	Batch size	Epoch	Top-1	Top-5
Supervised	None	256	90	76.5	93.0
AdCo [110]	MLP	256	800	72.8	-
AdCo [110]	MLP	1024	800	75.7	-
CaCo [111]	MLP	4096	800	75.7	-
CLSA [177]	MLP	256	800	76.2	-
DeepCluster-v2 [64]	MLP	256	400	74.3	-
DINO [142]	MLP	4096	300	75.3	-
MMCR [115]	MLP	256	100	72.1	-
NNCLR [61]	MLP	4096	1000	75.6	92.4
ReSSL [97]	MLP	4096	200	74.7	-
SeLa-v2 [64]	MLP	256	400	71.8	-
SMoG [140]	MLP	4096	400	**76.4**	93.1
SwAV [64]	MLP	4096	800	75.3	-
UniGrad [106]	MLP	4096	800	75.5	-

4.1.4 Comparative Results: Semi-supervised Evaluation on ImageNet-1K

Top-1 and top-5 accuracy are key evaluation metrics in machine learning classification tasks, and are used for performance comparison of SSL methods herein. Top-1 accuracy measures the percentage of test samples where the model's highest probability prediction matches the correct class, representing the conventional definition of accuracy. For example, classifying a cat image as "tiger" would be incorrect under top-1 accuracy. In contrast, top-5 accuracy measures the percentage of samples where the correct class appears among the model's five highest probability predictions, allowing for some flexibility in classification. This metric is particularly useful for problems with many classes, scenarios with multiple valid interpretations, recommendation systems, and complex datasets like ImageNet. Top-5 accuracy is always equal to or higher than top-1 accuracy, and the gap between these metrics can provide insights into model behaviour, potentially indicating that the model is learning relevant features but struggling with final discrimination between similar classes.

In the following, we gathered performance results of SSL methods in semi-supervised tasks using a ResNet-50 encoder for fair comparison. Table 4 presents the performance achieved when fine-tuning representations on classification tasks with small subsets (1% and 10%) of ImageNet-1K's training set. The results demonstrate that SimCLR-v2 (+KD) achieves the highest top-1 accuracy—73.9% with just 1% of labels and 91.5% with 10% of labels—significantly outperforming other methods. This superior performance can be attributed to the knowledge distillation technique employed in SimCLR-v2, though it should be noted that this approach is technically classified as semi-supervised rather than purely self-supervised. UniGrad follows closely behind, excelling with a top-5 accuracy of 83.8% using 1% of the labeled set and a top-1 accuracy of 71.5% with 10% of labels, respectively. NNCLR also demonstrates

Table 4: Performance comparison of SSL methods under semi-supervised fine-tuning protocol using limited labeled data from ImageNet-1K (1% and 10% of labels) with ResNet-50 backbone.

Method	1%		10%	
	Top-1	Top-5	Top-1	Top-5
Supervised	25.4	48.4	56.4	80.4
All4One [101]	39.0	-	60.1	-
Barlow Twins [68]	55.0	79.2	69.7	89.3
BYOL [29]	53.2	78.4	68.8	89.0
C-BYOL [86]	60.6	70.5	**83.4**	90.0
C-SimCLR [86]	51.1	77.2	64.5	86.5
InstDis [59]	-	39.2	-	77.4
MMCR (2 views) [115]	46.6	-	63.9	-
MMCR (8 views) [115]	51.0	-	67.7	-
MoCHI [93]	50.4	76.2	65.7	87.2
MSF [100]	55.5	79.9	66.5	87.6
NNCLR [61]	56.4	80.7	69.8	89.3
PCL [63]	-	75.3	-	85.6
PCL-v2 [63]	-	73.9	-	85.0
PIRL [77]	30.7	57.2	60.4	83.8
ReLIC-v2 [105]	58.1	72.4	81.3	91.2
SimCLR [28]	48.3	75.5	65.6	87.8
SimCLR-v2 (+KD) [82]	**73.9**	**91.5**	77.5	**93.4**
SMoG [140]	58.0	81.6	71.2	90.5
SwAV [64]	53.9	78.5	70.2	89.9
SynCo [93]	50.8	77.5	66.6	88.0
TiCo [125]	53.0	79.2	66.8	88.0
UniGrad [106]	60.8	83.8	71.5	90.6
UniVIP [131]	53.0	78.8	67.1	88.5
VICReg [160]	54.8	79.4	69.5	89.5

competitive performance, particularly when using 10% of labels, where it achieves a top-1 accuracy of 69.8%.

The comparison of SSL techniques across linear and semi-supervised evaluation frameworks reveals a consistent pattern in performance results. Methods that excel in linear evaluation generally demonstrate robust performance in semi-supervised settings as well, validating the effectiveness of their learned representations in label-constrained environments. For instance, ReLIC-v2, which leads in linear evaluation with a top-1 accuracy of 77.1%, also exhibits strong semi-supervised performance, achieving a notable top-1 accuracy of 81.3% with 10% of the labels. Similarly, SimCLR-v2, though technically classified as semi-supervised due to its knowledge distillation approach, outperforms other methods in its category, suggesting that strategies enhancing performance in linear evaluation settings transfer effectively to semi-supervised contexts. This correlation underscores the potential of SSL techniques not only in leveraging unlabeled data but also in efficiently utilizing small amounts of labeled data, presenting a compelling case for their broader applicative value in realistic machine learning

scenarios. Such consistency is crucial for validating the robustness of SSL methods and confirming their suitability across diverse real-world applications.

4.2 Transfer Learning Capabilities to Other Classification Tasks

Transfer learning evaluation differs from linear evaluation and semi-supervised fine-tuning on the source dataset by specifically measuring how well SSL-pretrained models generalize to new domains with semantically distinct data distributions and visual characteristics. Unlike linear evaluation on the source dataset, which freezes the encoder and trains only a classifier on the same dataset distribution, transfer learning requires adaptation to domain shifts through more extensive fine-tuning strategies, including domain-specific augmentations and architectural modifications. While evaluations on the source dataset have become relatively standardized, transfer learning protocols remain more fragmented, though efforts like [178] aim to standardize cross-domain evaluation. Furthermore, transfer learning evaluation more directly assesses real-world utility in specialized domains with scarce labeled data, revealing whether SSL methods learn truly generalizable features or merely dataset-specific patterns. Specifically, the features learned on ImageNet-1K are examined to determine their generality and usefulness across different image domains, and to assess whether they are inherently ImageNet-specific.

4.2.1 Evaluation Protocols

In transfer learning evaluation for computer vision SSL, the linear evaluation protocol assesses representation quality by freezing the pretrained encoder and training only a linear classifier on labeled target data. This process typically employs L-BFGS optimization with ℓ_2-regularization, as the convex nature of linear classification makes this approach efficient and stable. Training images undergo resizing to 256×256, random resized cropping to 224×224, horizontal flipping, and normalization, while validation images undergo resizing to 256×256, center cropping to 224×224, and normalization. The pipeline includes training with minimal hyperparameter tuning. This standardized approach enables direct comparison of representation quality across SSL methods while requiring minimal computational resources.

Semi-supervised fine-tuning evaluation, in contrast, updates some or all encoder weights alongside the classifier using limited labeled data and unlabeled target samples. This protocol employs SGD with Nesterov momentum (0.9) or AdamW (for ViTs or low-memory scenarios) with layer-wise learning rates, complemented by weight decay (ℓ_2-regularization) and consistency losses for pseudo-labeled data. The pipeline incorporates both weak augmentations (random crops/flips) and strong augmentations (e.g., RandAugment) with domain-specific adjustments, while the momentum parameter for batch normalization statistics is set to $\max(1 - 10/s, 0.9)$ where s denotes the number of steps per epoch. This approach achieves higher absolute performance than linear evaluation at the cost of increased computational requirements and hyperparameter sensitivity, making it more reflective of real-world deployment scenarios.

Table 5: Standard benchmark datasets used for evaluating transfer learning capabilities of self-supervised models on diverse classification tasks (following Kornblith et al. [171] evaluation protocol).

Dataset	Classes	Images (train/test)
Food-101 [179]	101	101,000 (75,750/25,250)
CIFAR-10 [173]	10	60,000 (50,000/10,000)
CIFAR-100 [173]	100	60,000 (50,000/10,000)
Birdsnap [180]	500	49,829 (47,386/2,443)
SUN397 [181]	397	39,700 (19,850/19,850)
StanfordCars [182]	196	16,185 (8,144/8,041)
FGVC Aircraft [183]	100	10,000 (6,667/3,333)
VOC 07 [184]	20	9,963 (5,011/4,952)
DTD [185]	47	5,640 (3,760/1,880)
Oxford-IIIT Pets [186]	37	7,049 (3,680/3,369)
Caltech-101 [187]	101	9,144 (3,060/6,084)
Oxford 102 Flowers [188]	102	8,189 (2,040/6,149)

4.2.2 Datasets

Transfer learning for classification is performed on the following datasets: Food-101 [179], CIFAR-10 [173] and CIFAR-100 [173], Birdsnap [180], the SUN397 scene dataset [181], Stanford Cars [182], FGVC Aircraft [183], the PASCAL VOC 2007 classification task [184], the Describable Textures Dataset (DTD) [185], Oxford-IIIT Pets [186], Caltech-101 [187], and Oxford 102 Flowers [188]. Table 5 presents details of these datasets. After linear evaluation or fine-tuning, performance is reported using standard metrics for each benchmark, and results are provided on a held-out test set after hyperparameter selection on a validation set. In the following subsection, we adopt the methodologies outlined in the respective dataset publications to report top-1 accuracy for Food-101, CIFAR-10, CIFAR-100, Birdsnap, SUN397, Stanford Cars, and DTD, and mean per-class accuracy for FGVC Aircraft, Oxford-IIIT Pets, Caltech-101, and Oxford 102 Flowers.

4.2.3 Comparative Results

In the following, we compiled transfer learning evaluation results for several SSL methods, all using a ResNet-50 encoder. Table 6 presents the results for various target datasets with the encoder frozen (linear evaluation), and Table 7 shows the results with the encoder fine-tuned.

Table 6 shows the performance of SSL methods under linear evaluation across several classification datasets. Supervised refers to models pre-trained on ImageNet using labeled data with direct class supervision. ReLIC-v2 stands out, achieving the highest accuracies on datasets such as Food-101 (80.6%), SUN397 (66.2%), and Aircraft (64.8%), indicating that ReLIC-v2 learns representations that generalize well to diverse image domains. NNCLR also performs exceptionally well, leading on CIFAR-10 (93.7%), CIFAR-100 (79.0%), and Birdsnap (61.4%). BYOL demonstrates strong

results on various datasets, with notable scores on Pets (90.4%) and Flowers (96.1%). These results highlight the effectiveness of ReLIC-v2, NNCLR, and BYOL in learning high-quality representations that perform well in linear evaluation.

Table 7 presents the results when the encoder is fine-tuned. Supervised learning refers to models pre-trained on ImageNet using labeled data with direct class supervision, serving as a strong baseline. Random init (initialization) refers to models with randomly initialized weights and no pre-training. TWIST achieves the highest accuracies on several datasets, including Food-101 (89.3%), CIFAR-10 (97.9%), and Birdsnap (76.7%). ReLIC-v2 also performs well, showing leading results on Food-101 (88.7%), Cars (92.3%), and Aircraft (88.7%). BYOL continues to demonstrate strong performance, particularly on Pets (91.7%) and Aircraft (88.1%). These observations indicate that TWIST, ReLIC-v2, and BYOL are highly effective in transfer learning tasks, achieving high accuracy across various datasets when fine-tuned.

The varying performance of SSL methods across vision tasks can be attributed to how different techniques capture and prioritize features during training. Methods like ReLIC-v2 and BYOL, which excel at capturing high-level semantic features, tend to perform well on diverse datasets requiring a broad understanding of object categories and visual concepts, such as Food-101 and Stanford Cars. On the other hand, methods like NNCLR, which may emphasize finer textural or shape details, excel on datasets like CIFAR-10 and Birdsnap, where such attributes are crucial for classification.

The performance variation also suggests that no single SSL approach is universally superior across all image domains, reflecting the specialized capabilities of different learning mechanisms. Datasets with fine-grained categories, like FGVC Aircraft and Oxford 102 Flowers, may benefit more from methods that capture intricate details and subtle within-category variations. Conversely, tasks requiring robustness to changes in background, lighting, or pose might favour methods good at generalizing from comprehensive scene understandings, such as those trained on SUN397 or VOC 2007.

4.3 Transfer Learning Capabilities to Other Vision Tasks

This evaluation assesses the generalization of representations beyond classification tasks, using various datasets and tasks, including but not limited to object detection, semantic segmentation, instance segmentation, and depth estimation. Object detection identifies and localizes objects in an image by drawing bounding boxes and assigning class labels. Semantic segmentation assigns a class label to every pixel in an image, creating a mask where all pixels of the same class share the same label. Instance segmentation, a subset of image segmentation, goes further by distinguishing individual instances of the same class, producing separate masks for each object. Depth estimation predicts the distance between the camera and points in a scene, generating a depth map that represents the 3D structure. The evaluation pipeline typically involves pretraining an encoder on unlabeled data (e.g., ImageNet-1K) using SSL objectives, integrating it into task-specific architectures (e.g., Faster R-CNN [117] for detection, DeepLabv3+ [189] for semantic segmentation), fine-tuning on labeled target datasets, and measuring performance using task-specific metrics like Average Precision (AP) for detection and mean Intersection over Union (mIoU) for semantic segmentation, or

Table 6: Transfer learning performance of various SSL methods on diverse classification datasets using features from ImageNet-1K pretrained ResNet-50 under linear evaluation protocol (frozen backbone).

Method	Food-101	CIFAR-10	CIFAR-100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Supervised	72.3	93.6	78.3	53.7	61.9	66.7	61.0	82.8	74.9	91.5	94.5	94.7
BYOL [29]	75.3	93.1	78.4	57.2	62.2	67.8	60.6	82.5	75.5	90.4	94.2	**96.1**
SimCLR [28]	68.4	90.6	71.6	37.4	**65.8**	50.3	50.3	80.5	74.5	83.6	90.3	91.2
NNCLR [61]	**76.7**	**93.7**	**79.0**	**61.4**	62.5	67.1	**64.1**	83.0	75.5	91.8	91.3	95.1
C-SimCLR [86]	73.0	91.6	75.2	38.2	62.3	52.7	53.5	-	73.0	84.0	91.2	89.0
ReLIC-v2 [105]	80.6	92.8	78.2	65.4	66.2	**75.1**	64.8	-	**77.4**	**92.4**	92.8	95.6
MSF [100]	71.2	92.6	76.3	-	59.2	55.6	53.7	-	73.2	88.7	92.7	92.0
TWIST [166]	91.2	74.4	66.8	-	66.8	55.2	53.6	**85.7**	76.6	91.6	**94.5**	93.4

Table 7: Transfer learning performance of various SSL methods on diverse classification datasets using fine-tuning of ImageNet-1K pretrained ResNet-50 models.

Method	Food-101	CIFAR-10	CIFAR-100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Supervised	88.3	97.5	86.4	75.8	64.3	92.1	86.0	85.0	74.6	92.1	93.3	97.6
Random init.	86.9	95.9	80.2	**76.1**	53.6	91.4	85.9	67.3	64.8	81.5	72.6	92.0
BYOL [29]	88.5	97.8	86.1	76.3	63.7	91.6	**88.1**	85.4	76.2	91.7	83.8	97.0
SimCLR [28]	88.2	97.7	85.9	75.9	63.5	91.3	88.1	84.1	73.2	89.2	92.1	97.0
MMCL [114]	82.4	96.2	82.1	-	-	89.2	85.4	-	73.5	-	87.8	95.2
ReLIC-v2 [105]	88.7	97.7	85.3	76.7	64.7	**92.3**	88.7	-	**76.9**	92.2	93.2	**97.9**
TWIST [166]	**89.3**	**97.9**	**86.5**	-	**67.4**	91.9	85.7	**86.5**	76.4	**94.5**	**93.5**	97.1

Table 8: Datasets used for evaluating transfer learning capabilities of self-supervised models on complex vision tasks beyond classification, including object detection, instance segmentation, semantic segmentation, and depth estimation.

Dataset	Task	Classes	Images (train/test)
VOC 07 [184]	Object detection	21	5,011 / 4,952
VOC 07+12 [184]	Object detection	21	16,551 / 4,952
COCO [132]	Object detection/Segmentation	80	118,287 / 5,000
LVIS v0.5 [192]	Instance segmentation	1,203	100,170 / -
ADE20K [193]	Semantic segmentation	150	20,210 / 2,000
VOC 2012 [184]	Semantic segmentation	21	11,530 / 1,449
Cityscapes [194]	Semantic segmentation	30	2,975 / 500
NYU v2 [195]	Depth estimation	-	24,000 / 654

Root Mean Square Error (RMSE) for depth estimation. Detailed evaluation protocols for each task, including target datasets and comparative results, are provided in the following subsections.

4.3.1 Evaluation Protocols and Datasets

In SSL transfer learning evaluation, ResNet-50 (hereby referred as R50) serves as the standard backbone architecture, offering a consistent framework for comparing various SSL methods across downstream tasks. This 50-layer deep convolutional neural network, featuring residual blocks and skip connections to address the vanishing gradient problem, provides a reliable basis for assessing representation quality. Performance metrics such as linear probing accuracy, fine-tuning results on object detection and segmentation, and transfer learning efficiency to different domains are commonly reported using R50 as the base encoder, enabling fair comparisons of SSL techniques. The widespread adoption of R50 in computer vision research, due to its robust design and ability to train very deep networks effectively, has solidified its position as a crucial benchmark in the field.

For object detection evaluation, the Faster R-CNN [117] detector (a two-stage object detection algorithm that uses a Region Proposal Network (RPN) and convolutional neural networks to identify and locate objects in images, providing bounding box predictions and class labels) with either frozen or fine-tuned representations, is employed. Backbone choices include R50-C4, R50-DC5, or R50-FPN, representing, respectively, the fourth stage output of ResNet-50, dilated convolutions at the fifth stage, and a feature pyramid network built on ResNet-50. This setup is implemented in Detectron2 [190], an open-source computer vision library providing a flexible framework for state-of-the-art object detection and segmentation, including pre-trained models and efficient training pipelines. Object detection performance is evaluated on the VOC 2012 [184] and COCO [132] datasets using Faster R-CNN for bounding box predictions and Mask R-CNN [191] for both object detection and instance segmentation, with Mask R-CNN extending Faster R-CNN by adding a branch for predicting pixel-level segmentation masks for each detected object.

For instance segmentation, the Mask R-CNN [191] detector with frozen or fine-tuned representations is used, with the same backbone options: R50-C4, R50-DC5, or R50-FPN. This setup is also implemented in Detectron2 [190]. Instance segmentation is assessed on the COCO [132], LVIS v0.5 [192], and Cityscapes [194] datasets, mirroring the approach in MoCo [16]. For semantic segmentation, a fully-convolutional network (FCN)-based [196] architecture as in MoCo [16] is used. The backbone consists of a pretrained ResNet-50. The FCN is trained using a standard per-pixel softmax CE loss. Semantic segmentation is evaluated on the VOC 2012 [184], Cityscapes [194], and ADE20K [193] datasets, using methodologies from prior research such as [196]. Finally, for depth estimation, the protocol from [197] is followed, utilizing a standard ResNet-50 backbone. The conv5 features are processed through four fast up-projection blocks with filter sizes of 512, 256, 128, and 64. Training is performed using a reverse Huber loss function, as specified in [197] and depth estimation is evaluated on the NYU v2 dataset [195]. Table 8 summarizes details of the aforementioned datasets.

4.3.2 Comparative Results

We collected performance evaluation results of several SSL methods on transfer learning tasks for object detection and instance segmentation using a ResNet-50 backbone for fair comparison. The results are summarized in Table 9. For detection tasks Faster R-CNN is employed, while Mask R-CNN is used for segmentation. AP (Average Precision), AP_{50}, and AP_{75} are key metrics for evaluating SSL in object detection/instance segmentation. AP is the mean average precision calculated over multiple Intersection over Union (IoU) thresholds, typically from 0.5 to 0.95 in 0.05 increments, providing a comprehensive assessment of detection/segmentation quality. AP_{50} specifically measures the average precision at an IoU threshold of 0.5 (50% overlap), offering a more lenient evaluation that's particularly useful for assessing performance on smaller or harder-to-detect objects. AP_{75}, with its higher IoU threshold of 0.75, demands more precise localization and is thus a stricter measure of accuracy. These metrics collectively offer a nuanced view of an SSL model's ability to transfer learned representations to detection and instance segmentation tasks, balancing between coarse object localization (AP_{50}) and more precise boundary delineation (AP_{75}), with AP providing an overall performance summary.

Among the evaluated methods, SoCo and InsCon achieve state-of-the-art performance on both VOC and COCO benchmarks. On VOC detection, InsCon leads with the highest scores across all metrics (AP = 59.1, AP_{50} = 83.6, AP_{75} = 66.6), demonstrating its effectiveness in object recognition and boundary localization. For COCO detection, SoCo outperforms all methods with AP^{bb} = 40.4, AP_{50}^{bb} = 60.4, and AP_{75}^{bb} = 43.7, while InsCon follows closely in second place. On COCO segmentation, InsCon achieves the top performance with AP^{msk} = 35.1 and AP_{75}^{msk} = 37.6, while SoCo leads in AP_{50}^{msk} = 56.8. These results highlight how methods focusing on object-level representations (SoCo) and multi-instance consistency (InsCon) significantly outperform both supervised baselines and earlier self-supervised approaches.

Table 9: Transfer learning performance of self-supervised methods on object detection and instance segmentation tasks using ResNet-50-C4 backbone. Results are reported on PASCAL VOC for detection and COCO for both detection and segmentation using standard Average Precision metrics (AP, AP_{50}, AP_{75}). Detection metrics are denoted by bb (bounding box) and segmentation by msk (mask). All methods used 200-epoch pretraining except where noted in parentheses.

Method	VOC detection			COCO detection			COCO segmentation		
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{msk}	AP^{msk}_{50}	AP^{msk}_{75}
Supervised	53.5	81.3	58.8	38.2	58.2	41.2	33.3	54.7	35.2
Random init	33.8	60.2	33.1	26.4	44.0	27.8	29.3	46.9	30.8
Barlow Twins [68]	56.8	82.6	63.4	39.2	59.0	42.5	34.3	56.0	36.5
BYOL [29]	51.9	81.0	56.5	-	-	-	-	-	-
DenseCL [118]	58.7	82.8	65.2	-	-	-	-	-	-
DetCo [129]	57.8	82.6	64.2	39.8	59.7	43.0	34.7	56.3	36.7
InfoMin [76]	57.6	82.7	64.6	39.0	58.5	42.0	34.1	55.2	36.3
InsCon [130]	**59.1**	**83.6**	**66.6**	40.3	60.0	43.5	**35.1**	56.7	**37.6**
InsLoc [123]	57.9	82.9	64.9	39.5	59.1	42.7	34.5	56.0	36.8
InstDis [59]	55.2	80.9	61.2	37.7	57.0	40.9	33.0	54.1	35.2
MoCHI [92]	57.5	82.7	64.4	39.2	58.9	42.4	34.3	55.5	36.6
MoCo [16]	55.9	81.5	62.6	38.5	58.3	41.6	33.6	54.8	35.6
MoCo-v2 [88]	57.0	82.4	63.6	38.9	58.6	41.9	34.1	55.5	36.0
PIRL [77]	55.5	81.0	61.3	37.4	56.5	40.2	32.7	53.4	34.7
ReSim [122]	58.7	83.1	66.3	39.7	59.0	43.0	34.6	55.9	37.1
SimCLR [28]	56.3	81.9	62.5	-	-	-	-	-	-
SimSiam [53]	57.0	82.4	63.7	39.2	59.3	42.1	34.4	56.0	36.7
SoCo [128]	**59.1**	83.4	65.6	**40.4**	**60.4**	**43.7**	34.9	**56.8**	37.0
SwAV [64]	56.1	82.6	62.7	38.4	58.6	41.3	33.8	55.2	35.9
SynCo [93]	57.2	82.6	63.9	39.9	59.6	43.3	34.9	56.5	36.9

5 Discussion

This section delves into a comprehensive discussion of fundamental challenges inherent in discriminative SSL and their impact on the efficacy of learned representations. We begin with an analysis of the architectural evolution of SSL frameworks (Section 5.1), exploring how base networks, pretext tasks, and training objectives have transformed over time. Next, we examine the fundamental challenges facing discriminative SSL methods (Section 5.2), including representation collapse, dimensionality reduction, and hyperparameter sensitivity. We then address the intrinsic trade-offs encountered within SSL (Section 5.3), exploring the tensions between seemingly desirable properties such as invariance-equivariance and practical limitations of scalability and efficiency. The discussion continues with critical considerations for benchmarking and evaluation protocols (Section 5.4), including limitations of current benchmarks and assessment of

robustness. Finally, we examine the theoretical foundations underpinning SSL methods (Section 5.5), highlighting both formal guarantees and gaps in our understanding. The primary objective of these in-depth analyses is to foster a deeper understanding for the reader, identify prevailing trends, and propose promising avenues for future research.

5.1 Evolution of Architectural Frameworks

The evolution of SSL methods in computer vision has shown significant progress in recent years. As shown in Figure 12, there is a clear upward trend in top-1 accuracy on ImageNet-1K, showing the steady improvement of SSL techniques. This progress reflects the field's development, with recent methods like ReLIC-v2 even surpassing the supervised learning baseline of 76.5%, an important milestone that highlights the potential of learning without labels. Examining contrastive learning frameworks reveals a clear pattern of development. Early methods like MoCo used simple projection heads, while later approaches such as SimCLR and MoCo-v3 used more complex MLPs with more hidden layers. The use of asymmetric architectures, introduced by BYOL, has also been very effective. This trend suggests that architectural complexity correlates with performance improvements, following a *"bigger is better"* principle.

This upward trend shifts after 2023, where the top-1 accuracy begins to decrease slightly. This change occurs because newer methods focus on different objectives rather than just improving ImageNet accuracy. Recent approaches address specific limitations in previous methods (e.g., SynCo), combine SSL techniques with transformer architectures (e.g., All4One), or optimize for specific downstream tasks beyond classification (e.g., VICRegL for object detection). These developments show how the field is expanding beyond benchmark performance to solve more practical problems.

However, current frameworks appear to be approaching a performance bottleneck, indicating the need to revisit their fundamental limitations rather than simply scaling them further. For example, a significant challenge in contrastive learning is the diminishing contribution of negative samples as training progresses [92, 93]. Several approaches have attempted to address this limitation, including methods like MoCHI and SynCo that generate synthetic hard negatives, and DCL that modifies the InfoNCE loss to decouple negative and positive pairs. This highlights the importance of identifying and addressing specific weaknesses in existing frameworks rather than solely developing entirely new paradigms. In this section, we summarize the architectural framework evolution of SSL methods in terms of the base network, the shift in pretext tasks, and the evolution of training objectives and loss functions, and discuss how each of these aspects contributes to the advancement of the field.

5.1.1 Base Network Evolution

While ResNet-50 serves as a consistent baseline for fair comparisons, the pursuit of better performance has led to the adoption of more advanced encoder architectures like larger ResNet variants and Vision Transformers, actively exploring and adopting advanced architectures to enhance SSL capabilities. More specifically, the evolution of base networks in discriminative SSL progressed from ResNet-dominated frameworks

Fig. 12: Evolution of SSL performance on ImageNet-1K linear evaluation using ResNet-50 encoder across different methods and years (2017-2025).

(2018–2020)—where methods like MoCo (2019), SimCLR (2020), and SwAV (2020) leveraged ResNet-50's hierarchical features for contrastive and clustering tasks—to intermediate architectures like Wide ResNets and deeper hybrids (e.g., ResNet-50 ×2 in MoCo-v2), which enhanced feature capacity and stability. ReLIC-v2 (2022) further optimized ResNet backbones (up to ResNet-200) with multi-view invariance and saliency masking, achieving 80.6% ImageNet-1K accuracy and surpassing supervised baselines, demonstrating that advanced SSL objectives could revitalize CNNs even as ViTs gained prominence.

Following the introduction of ViTs, hierarchical architectures like EsViT (2022) integrated CNN-like multi-stage processing with self-attention, achieving 81.3% accuracy. While lightweight CNNs prioritized efficiency, ViTs, exemplified by DINO (2021) with its ViT-Small and self-distillation (78.2% accuracy), demonstrated superior performance post-2021. Barlow Twins (2021) also refined ResNet training through projection heads, further illustrating the evolving landscape. However, the direct adaptation of ViT architectures for SSL proved challenging, necessitating specific modifications for stable training. For instance, MoCo-v3 freezes the initial patch embedding layer to mitigate gradient spikes, while MoBY employs asymmetric drop path rates for its encoders. The distinct global dependency capture of ViTs' self-attention, compared to CNNs' hierarchical feature extraction, suggests the potential for novel self-supervision forms. These challenges underscore the need for SSL techniques tailored to the unique characteristics of transformer architectures, rather than mere adaptations of CNN-based methods. Conversely, the success of ReLIC-v2 highlighted that architectural refinements and training protocols, such as saliency-aware augmentation, can bridge the ResNet-ViT performance gap, providing a robust, CNN-compatible alternative to ViT-driven methods like DINO. This trajectory reflects a

63

dual focus: scaling ViTs for global context capture while optimizing CNNs through SSL innovations, ensuring the continued viability of both architectures in discriminative SSL.

5.1.2 Shift in Pretext Tasks

The evolution of pretext tasks in SSL for computer vision transitioned from diverse early approaches (including colorization [43], inpainting [44], jigsaw puzzles [10], and rotation prediction [46]) to data augmentation-dominated frameworks, reflecting the field's prioritization of invariance-driven objectives. Early pretext tasks required models to solve spatial-contextual challenges (e.g., predicting relative patch positions or restoring color to grayscale images), which fostered rich but task-specific representations. However, the rise of contrastive learning (e.g., SimCLR, MoCo) shifted focus to augmentation-based pretext tasks, where models learn invariance to transformations like cropping, flipping, or color jittering [28]. While this shift improved scalability and alignment with downstream tasks, it risked narrowing the diversity of learned features by over-relying on augmentation-induced invariance. Despite this trend, methods like CPC [21], which leverage temporal or spatial predictive coding, remain relevant for capturing complementary structural patterns, underscoring the importance of maintaining methodological diversity in SSL. This progression highlights a tension between standardization for efficiency and the preservation of task-agnostic representation breadth.

Recent advancements (2023–present) introduced automated pretext task design, exemplified by evolutionary augmentation policies (e.g., EAPO [198]) optimizing data transformations via genetic algorithms, and multi-task hybrids combining contrastive learning with spatial reasoning (e.g., rotation and contrastive objectives). These innovations improved performance across domains: PIRL boosted PASCAL VOC detection by 9%, [198] elevated CIFAR-10 accuracy by 12%, and hybrid tasks enhanced ImageNet linear probing by 7%. Applications expanded to medical imaging (e.g., patch reconstruction for tumor detection) and video analysis (frame-order prediction for action recognition), reflecting a shift from manual heuristic design to systematic, optimization-driven approaches that rival supervised learning in label-scarce settings.

5.1.3 Evolution of Training Objectives and Loss Functions

The evolution of training objectives in discriminative SSL reflects a deliberate shift from heuristic designs to mathematically grounded principles. Early contrastive methods like SimCLR relied on NCE, which explicitly compared positive pairs against negatives to maximize mutual information. While effective, this approach faced scalability challenges due to its dependence on large negative sample batches and sensitivity to augmentation strategies. This limitation spurred the development of non-contrastive paradigms like BYOL, which eliminated negatives entirely by enforcing consistency between a student network and an EMA teacher. The success of such methods revealed that feature collapse could be prevented through architectural mechanisms (e.g., predictor networks) rather than explicit contrastive terms.

Concurrently, feature decorrelation objectives emerged as a unifying framework across families. Methods like VICReg and Barlow Twins replaced instance discrimination with covariance matrix constraints, explicitly minimizing redundancy between feature dimensions. This shift aligned with theoretical insights showing that dimensional collapse – not just complete collapse – hindered representation quality. Hybrid approaches soon followed: DINO combined self-distillation with centering and sharpening operations to stabilize training, while MSN integrated MIM with clustering objectives. These innovations demonstrated that multi-objective loss functions could balance invariance (to augmentations) with variance (across semantic content), addressing earlier trade-offs between representation robustness and discriminability.

Future advancements in training objectives will likely focus on integrating causal reasoning and task-adaptive dynamics to bridge the gap between pretraining and downstream applications. Inspired by insights from dimensional collapse analysis, next-generation loss functions may explicitly model causal feature disentanglement, penalizing spurious correlations induced by augmentation strategies while preserving invariant semantic factors. For instance, objectives could incorporate counterfactual terms that simulate interventions on nuisance variables (e.g., lighting, occlusions), fostering more robust feature spaces. Simultaneously, meta-learned loss functions might emerge, where the objective itself is optimized during training to balance competing goals like invariance, decorrelation, and task-specific discriminability. This could build on frameworks like hypernetworks or gradient-based meta-learning, dynamically adapting loss landscapes to dataset statistics. Additionally, scalable multi-objective hybrids may unify contrastive, distillation, and decorrelation mechanisms through learnable weightings, avoiding the manual tuning seen in current hybrid SSL pipelines. Such systems could automatically prioritize feature diversity in early training before shifting to semantic alignment, mimicking curriculum learning principles. These directions aim to move beyond static, handcrafted losses toward adaptive, theoretically coherent objectives that generalize across domains.

5.1.4 Beyond Contrastive and Clustering Methods

While contrastive learning and clustering approaches have largely defined the SSL landscape, other methods deserve more attention. Self-distillation techniques, such as BYOL and SimSiam, and feature decorrelation methods, including Barlow Twins, VICReg, and ReLIC, offer useful alternatives that have not been explored as much. In particular, regularization-based methods like VICReg and ReLIC show promise for future work. This aligns with ideas from Yann LeCun, who has suggested focusing more on regularization techniques in representation learning. However, contrastive methods often suffer from a "semantic repulsion" problem [150], where semantically similar instances are forced apart even when they should maintain similarity. These regularization approaches could address this issue, along with other limitations of contrastive methods, such as the need for many negative samples or the risk of losing valuable information. By setting direct constraints on the statistical properties of representations, they may improve robustness and generalizability. Future research should examine these regularization methods closely, aiming to create new objectives that

ensure invariance to irrelevant transformations while keeping information relevant to downstream tasks.

5.2 Fundamental Challenges in Discriminative Self-supervised Learning

The preceding sections highlighted that SSL methods, while demonstrably successful, are susceptible to fundamental challenges that can impede their efficacy and the fidelity of learned representations. These challenges include representation collapse, dimensionality collapse, semantic repulsion, early training degeneration, and acute hyperparameter sensitivity. A thorough understanding of the interrelationships among these issues and their manifestation across diverse SSL methods is essential for advancing the field and developing robust, reliable techniques. This subsection provides a comprehensive overview of these interrelationships, identifying the specific vulnerabilities of each SSL method family and exploring potential inherent immunities or effective mitigation strategies.

5.2.1 Representation Collapse

Representation collapse, also known as system collapse, represents a critical failure mode in SSL where the learned representations become trivial and fail to capture any meaningful information from the input data [87, 168]. This often manifests as the model mapping all input instances to the same constant vector or to a very limited subspace of the representation space. Such a scenario renders the learned representations unusable for downstream tasks because they lack the discriminative power to distinguish between different inputs. One of the primary causes of representation collapse is a poorly designed loss function that does not incorporate sufficient constraints to prevent the model from converging to these trivial solutions. For instance, if the objective is solely to maximize the similarity between different augmented views of the same input, without any mechanism to ensure diversity in the representations of different inputs, the model might simply learn to output the same constant vector for all inputs, thereby achieving maximal similarity within positive pairs but losing all discriminative information. Representation collapse can be considered an extreme form of dimensionality collapse, where the representation space effectively shrinks to a single point or a very low-dimensional space. Most SSL methods incorporate specific design elements to avoid this fundamental failure, but the strategies employed can sometimes lead to other challenges, such as dimensionality collapse in a less extreme form or an over-reliance on certain aspects of the training process.

The field has developed a diverse array of techniques to mitigate this degenerate solution. For example, methods like DINO employ sharpening and centering mechanisms, BYOL and MoCo utilize momentum encoders to maintain a slowly evolving representation of past states, while SwAV introduces swapping assignments between views. Feature decorrelation methods like VICReg and ReLIC implement explicit regularization terms to ensure diverse representations.

5.2.2 Dimensionality Collapse

Dimensionality collapse describes a scenario where learned representations, though not entirely constant, reside within a lower-dimensional subspace of the model's embedding space, occurring when only a few dimensions capture significant data variance, leaving many others unused or redundant, often evidenced by a covariance matrix with dominant eigenvalues indicating variance concentration along few principal components. This collapse restricts the model's ability to encode comprehensive data information, potentially hindering performance on tasks requiring detailed distinctions, and can manifest not only in final representations but also in hidden features and encoder weight matrices, where it leads to self-related, redundant filters that limit network expressiveness. However, a degree of dimensionality reduction can be beneficial for separability, as models naturally focus on discriminative features, and recent findings suggest it might be an emergent property for informative representation learning; thus, the objective is not to eliminate reduction entirely, but to ensure the resulting subspace maintains sufficient information for downstream tasks.

Various techniques have been proposed to mitigate dimensionality collapse. Direct-CLR addresses dimensional collapse by eliminating the trainable projector and directly optimizing a subvector of the representation space, preserving the full dimensionality of the embeddings. OR applies orthogonality regularization, enforcing orthogonal weight matrices to prevent dimensional collapse across weights, hidden features, and representations, ensuring filter diversity and uniform eigenvalue distributions, targeting both convolutional and linear layers during pretraining. By ensuring that the weight matrices are orthogonal, OR helps to stabilize the distribution of their outputs, thereby preventing the dimensional collapse of hidden features and final representations. Feature whitening (used in Barlow Twins and VICReg) is another technique that aims to decorrelate the features. However, some evidence suggests that feature whitening might only address the issue at the feature level and might not prevent or could even exacerbate collapse in the weight matrices. This diversity of approaches reflects the challenge's significance and the field's ingenuity in addressing it. The necessity of these "tricks" underscores that representation collapse remains a fundamental theoretical issue in SSL that warrants continued attention.

5.2.3 Semantic Repulsion

Semantic repulsion is a counterintuitive outcome where learned representations, instead of grouping semantically similar instances, push them apart, a phenomenon arising from imbalances or flaws in the learning objective or training process, particularly in contrastive and clustering-based methods intended to do the opposite. Potential causes include overly aggressive negative sampling that treats semantically related instances as negative pairs, or poor initial cluster assignments and refinement in clustering methods that misclassify similar instances, leading to distant representations. This repulsion detrimentally affects representation quality, hindering downstream tasks reliant on semantic similarity, such as image retrieval and semantic classification, by distorting the representation space's reflection of underlying semantic relationships; mitigating this requires careful design of SSL objectives to balance

attraction and repulsion of instances, and robust mechanisms for identifying and grouping semantically related data points.

5.2.4 Early Degeneration

Early degeneration describes the premature plateauing or decline of model performance, measured by downstream task representation quality, hindering the model's ability to learn optimal, generalizable representations. This issue arises from factors like convergence to suboptimal local minima, overfitting to pretext tasks, and problematic training dynamics, such as imbalances in positive and negative signals or optimization flaws, including the uniformity-tolerance dilemma in contrastive learning or teacher model stagnation in self-distillation. Consequently, learned representations become less informative and robust, leading to diminished downstream task performance compared to potential outcomes with extended or stable training; mitigating early degeneration necessitates meticulous hyperparameter tuning, exploration of alternative optimization strategies, and adjustments to SSL objectives to promote sustained learning and improved generalization.

5.2.5 Hyperparameter Sensitivity

Data augmentations are fundamental to discriminative SSL, enabling models to learn invariant features and generalize by creating multiple transformed views of input data, thereby focusing on essential semantics; however, SSL performance is highly sensitive to the choice, type, and strength of these augmentations, where appropriate augmentations significantly improve performance while poorly chosen or overly aggressive ones can degrade learned representations by losing critical information or learning spurious invariances, and conversely, weak augmentations may fail to provide sufficient diversity, leading to trivial feature learning. This sensitivity varies across SSL families, with contrastive learning being particularly susceptible due to its reliance on positive and negative pair creation, and recent research challenging the label-preserving assumption by exploring label-destroying augmentations as feature dropout. Notably, large datasets may reduce the necessity for complex augmentations, revealing a complex interaction between data scale and augmentation strategies; consequently, there is growing interest in adaptive selection or learning of optimal augmentations tailored to specific tasks and datasets to mitigate this sensitivity.

5.2.6 Addressing the Challenges

Contrastive methods excel in self-supervised representation learning by discerning between similar and dissimilar data pairs, mitigating complete representation collapse through loss functions that maximize inter-instance distance, often reliant on numerous negative samples. Despite this, dimensionality collapse remains a concern, necessitating techniques like projector networks, feature decorrelation, or orthogonality constraints. Furthermore, the definition of negative pairs can induce semantic repulsion, pushing semantically related instances apart. Early degeneration may stem from the uniformity-tolerance dilemma or suboptimal negative sample selection. Finally,

performance is highly sensitive to augmentation choices and hyperparameters, such as temperature and learning rate.

Clustering-based SSL groups similar data points, avoiding collapse via overclustering or multi-view prediction tasks. Although the clustering objective can encourage dimensional usage, dimensionality collapse remains a possibility. Semantic repulsion is less likely when clustering accurately reflects semantic similarity, but poor initialization or training instability can separate related instances. Early degeneration is a risk if clustering converges prematurely. The method's effectiveness hinges on hyperparameters related to the clustering algorithm and the overall training process.

Self-distillation, exemplified by BYOL and SimSiam, trains a student network to predict a teacher's output, eschewing negative samples and relying on asymmetric architectures and momentum encoders to prevent collapse. While dimensionality collapse remains a potential issue, the prediction task can promote dimensional utilization. Semantic coherence in the teacher's representations can mitigate semantic repulsion, though early degeneration can occur if the teacher stagnates or the student struggles to learn. Performance is contingent upon hyperparameters, particularly learning rate and momentum.

Knowledge distillation in SSL transfers knowledge from a teacher to a student, leveraging the teacher's collapse-avoidance capabilities. The student's susceptibility to dimensionality collapse is influenced by the teacher's representation space. Semantic repulsion can be inherited from the teacher or arise during student training. Early degeneration in the student can result from ineffective knowledge transfer or limited capacity. Both teacher and student training, as well as the distillation process, are sensitive to hyperparameter tuning.

Feature decorrelation methods explicitly combat collapse and dimensionality reduction by promoting statistical independence among representation dimensions, often via loss function penalties. While effective against collapse, care must be taken to avoid inadvertently inducing semantic repulsion. Early degeneration remains a concern, requiring robust training strategies. Performance is influenced by hyperparameters related to decorrelation strength and broader training parameters.

5.3 Trade-offs

In the following we delve into the inherent trade-offs encountered in SSL, exploring the fundamental tensions between seemingly desirable properties and practical limitations. We begin by examining the invariance-equivariance spectrum in representation learning, followed by a discussion on the interplay between scalability and efficiency in SSL methodologies. Finally, we address the trade-offs that arise when optimizing SSL methods for specific downstream tasks.

5.3.1 Invariance-Equivariance Trade-off

In the realm of representation learning, particularly within SSL, there exists a fundamental trade-off between learning representations that are invariant to certain transformations and those that are equivariant to them. Invariance refers to the property where a representation remains unchanged despite the input undergoing a specific

transformation, such as rotation or cropping. This is often desirable for high-level tasks like image classification, where the category of an object should be recognized regardless of its orientation or position. Equivariance, on the other hand, means that the representation transforms in a predictable way that corresponds to the transformation applied to the input. This property is often more beneficial for tasks that require spatial awareness, such as object detection or semantic segmentation, where the location and spatial relationships of objects are crucial.

Different families of SSL methods handle this trade-off in various ways. Contrastive learning, for example, often aims to learn representations that are invariant to the data augmentations used during training. By encouraging different augmented views of the same image to have similar representations, contrastive methods implicitly promote invariance to these transformations. However, this pursuit of invariance can sometimes lead to the loss of information that might be valuable for certain downstream tasks. Recognizing this limitation, recent research has explored ways to incorporate equivariance into SSL frameworks, often resulting in methods termed "contrastive-equivariant self-supervised learning". These approaches aim to learn representations that are both robust to certain variations (invariant) while also preserving information about specific transformations (equivariant).

Clustering-based SSL methods typically aim for invariance within the learned clusters, meaning that instances belonging to the same semantic cluster should have similar representations regardless of certain transformations. However, the definition of these clusters and the features used for clustering can influence the degree of invariance or equivariance captured. Feature decorrelation methods often focus on learning representations where the different dimensions are statistically independent, which can implicitly affect the invariance or equivariance properties of the learned features. Recent research continues to explore methods that can learn both invariant and equivariant features simultaneously or adaptively, based on the requirements of the specific downstream task. This highlights the ongoing effort to find the right balance in the invariance-equivariance spectrum for different applications of SSL.

5.3.2 Scalability and Efficiency

Scalability and efficiency in discriminative SSL methods exhibit both complementary dynamics and inherent trade-offs, shaped by architectural choices and optimization strategies. Scalability—the ability to maintain performance gains as model size or data volume increases—often conflicts with efficiency, which prioritizes reduced computational costs (e.g., memory, training time). Early contrastive methods like SimCLR achieved scalability through large batch sizes but suffered inefficiency due to quadratic memory costs from negative samples. This spurred innovations like MoCo's memory-efficient queue-based negatives and BYOL's elimination of negatives via self-distillation, decoupling scalability from batch size constraints. ViTs further altered this balance: while their self-attention mechanisms scale poorly with input resolution, techniques like patch-based contrastive learning and gradient checkpointing enabled efficient training on high-resolution images. Recent trends show a shift toward non-contrastive paradigms (e.g., VICReg, DINO) that avoid explicit negative comparisons,

achieving scalability through feature decorrelation or self-distillation objectives while maintaining sublinear computational growth.

Families differ markedly in their scalability-efficiency profiles. Contrastive methods remain computationally heavy for large datasets but benefit from parallelizable architectures. Self-distillation (e.g., DINO) and feature decorrelation families (e.g., Barlow Twins) excel in efficiency by eliminating negative samples and simplifying loss calculations, though they may require longer training cycles to stabilize. Clustering-based methods like SwAV balance both dimensions through online cluster assignments but face diminishing returns on extremely heterogeneous data. The future lies in adaptive hybrid frameworks: methods like MSN combine masked image modeling with clustering to dynamically allocate compute resources. As shown in DINO-v2's scaling to 7B parameters, ViT-based SSL models will likely dominate, leveraging sparse attention mechanisms and mixed-precision training to reconcile scalability with manageable compute budgets

5.3.3 Trade-offs in Task-Specific Optimization

Recent years have witnessed an increasing focus on task-specific SSL methods, particularly for dense prediction tasks like object detection and segmentation. Methods such as DenseCL, InsLoc, and SoCo have achieved strong results on these tasks, often outperforming supervised pretraining. However, this specialization comes with trade-offs. As evidenced by methods like ReLIC-v2 and DINO-v2, optimizing for dense representation learning can marginally reduce classification accuracy while significantly boosting performance on localization tasks. This highlights a fundamental tension in SSL: learning local features often comes at the expense of global features, creating a situation where improving performance on one downstream task may degrade performance on another. Future research should seek to develop methods that better balance these competing objectives.

5.4 Benchmarking and Evaluation

Evaluation protocols for discriminative SSL in computer vision have shifted from simplistic in-domain metrics (e.g., linear probing on ImageNet) to comprehensive frameworks assessing generalization, robustness, and architectural adaptability. Early protocols prioritized linear separability but overlooked hyperparameter sensitivity and domain shifts. Modern benchmarks now evaluate out-of-distribution performance across diverse datasets, quantify invariance-variance trade-offs, and measure few-shot adaptability. Standardization efforts introduced input normalization (e.g., batch norm before linear evaluation) and stratified hyperparameter testing to reduce inconsistencies between SSL families. Evaluations increasingly prioritize real-world viability, testing on niche industrial tasks (e.g., medical imaging) and isolating backbone contributions (e.g., ViTs vs. CNNs). Future directions emphasize multi-modal zero-shot transfer (e.g., CLIP-style alignment) and causal robustness to inference-time perturbations, reflecting a maturation toward practical applicability over narrow benchmark optimization.

The subsequent discussion will focus on critical considerations pertaining to the evaluation and practical application of SSL methods. We delve into the limitations of current benchmarking practices and the crucial aspect of data diversity, followed by an analysis of influential SSL frameworks. We then address the often-overlooked dimensions of robustness and out-of-distribution generalization, provide a brief comparison with the burgeoning field of masked image modeling, and conclude by discussing the significant practical constraints related to computational accessibility.

5.4.1 Benchmark Limitations and Data Diversity

The reliance on ImageNet as the primary benchmark for SSL raises important questions about the field's direction. While ImageNet provides a standardized evaluation platform, it is a meticulously curated, balanced dataset that may not reflect the challenges of learning from truly uncurated unblanced data. The aspiration of SSL is to leverage vast amounts of unfiltered, imbalanced, and potentially noisy data, yet most methods are demonstrated on carefully selected datasets. Future research should explore benchmarks that better reflect real-world data distributions, including significant class imbalance, noisy labels, and greater domain diversity. This would provide a more realistic assessment of SSL methods' ability to extract useful information from imperfect data sources. Additionally, transparency in pretraining datasets is essential for fair comparison, as some commercial entities utilize private datasets that cannot be replicated by the broader research community.

5.4.2 Evaluating Self-supervised Learning Frameworks

When evaluating *"what is the best self-supervised learning framework?"*, no single answer emerges. Different methods perform well in different contexts: some achieve superior classification performance, while others demonstrate strengths in detection or segmentation tasks. However, in terms of practical impact and adoption, MoCo-v2, SwAV, and DINO stand out as particularly influential frameworks for representation learning. MoCo introduced key innovations in contrastive learning, SwAV introduced effective clustering-based approaches, and DINO demonstrated strong performance with emerging properties like unsupervised segmentation. These frameworks have been widely adapted for domain-specific applications, including medical imaging and related fields.

5.4.3 Robustness and Out-of-Distribution Evaluation

A notable gap in current SSL evaluation protocols is the limited assessment of robustness and out-of-distribution (OOD) generalization. The primary objective of SSL is to create representations from large, uncurated datasets that generalize well to diverse scenarios. However, most evaluations focus on performance on well-curated, balanced datasets like ImageNet. This disconnect between the stated goal and evaluation practice represents a significant limitation in current research. Only a few methods, such as SynCo, explicitly address robustness to distribution shifts and adversarial attacks. A more systematic evaluation framework that includes diverse OOD datasets and

robustness metrics would provide a more comprehensive assessment of SSL methods' real-world utility. This would also help determine whether learned representations are truly generic or merely domain-specific.

5.4.4 Comparative Analysis with Masked Image Modeling

Although beyond the scope of this review, it is worth briefly discussing the relationship between discriminative SSL methods and generative approaches like MIM. Methods like BEiT [47] and Masked Autoencoders (MAEs) [199] have gained prominence, particularly with ViT architectures. While generative approaches typically require end-to-end fine-tuning for downstream tasks, discriminative SSL methods often perform well with linear probing. Generative methods like MAE can be more computationally efficient during pretraining (requiring only one forward pass instead of two), but their linear probing performance generally lags behind contrastive methods. Combining contrastive objectives with generative approaches (as in SimMIM [200]) can improve performance but at significant computational cost. This suggests complementary strengths between these paradigms that future research might productively integrate.

5.4.5 Practical Constraints and Accessibility

A significant practical challenge in advancing SSL research is the computational burden imposed by many state-of-the-art methods. Methods like SimCLR, MoCo-v3, Barlow Twins, and VICReg typically require large batch sizes (often 4096 or more) for optimal performance. Methods employing multi-crop augmentation, such as SwAV and SMoG, further increase computational demands. These requirements make SSL research inaccessible to many academic labs and smaller organizations lacking high-end computing infrastructure. Addressing this accessibility gap represents an important research direction. Methods like MoBY that demonstrate competitive performance with smaller batch sizes offer promising examples. Developing more computationally efficient SSL approaches would democratize research in this field and potentially lead to more diverse and innovative contributions from a broader community of researchers.

5.5 Theoretical Foundations

Most discriminative SSL methods are underpinned by various theoretical interpretations, although formal guarantees are still evolving. Contrastive learning and feature decorrelation methods stand out for their robust theoretical foundations, rooted in spectral graph theory, mutual information maximization, and information bottleneck principles, providing guarantees for linear separability and redundancy reduction through techniques like covariance matrix standardization; clustering-based approaches, while less formally grounded, leverage manifold learning to create transferable clusters aligned with data structures, using pseudo-labels for training; self-distillation methods, such as BYOL and DINO, focus on preventing representation collapse through momentum encoders and specific loss components, with emerging theories suggesting implicit decorrelation and mutual information maximization between

teacher-student models, despite lacking formal performance guarantees; knowledge distillation, though sharing student-teacher concepts, is primarily a supervised technique for model compression or transfer, not an SSL pre-training method; overall, SSL methods vary from rigorous, theory-backed approaches like contrastive and decorrelation, to heuristic-driven but empirically effective methods like clustering and distillation, reflecting a spectrum from spectral and information-theoretic principles to practical, semantic manifold or label consistency-focused strategies.

Regarding performance guarantees, theoretical work primarily provides assurances for linear probe performance under specific assumptions about data distributions and augmentations, suggesting the learned features can be linearly separable for downstream tasks. However, providing rigorous guarantees for the more common and complex scenario of fine-tuning deep networks on diverse downstream tasks remains a significant challenge due to the non-linearities involved. Much of the theoretical analysis for methods like self-distillation focuses on proving the avoidance of representation collapse rather than direct downstream performance. Thus, while theoretical foundations offer valuable insights into why these methods work, their widespread adoption is largely driven by strong empirical results, with ongoing research seeking to bridge the gap between theory and practice, especially concerning robust performance guarantees.

6 Conclusion

This review has provided a comprehensive analysis of discriminative SSL methods in computer vision, categorizing them into contrastive methods, clustering methods, self-distillation methods, knowledge distillation methods, and feature decorrelation methods. Through extensive evaluation and comparison, we have traced the evolution of these approaches and identified key trends, challenges, and opportunities in the field.

The progress in SSL has been substantial, with recent methods even surpassing supervised learning baselines on standard benchmarks. This success demonstrates the potential of learning from unlabeled data, a capability with significant implications for domains where labeled data is scarce or expensive to obtain. However, several important challenges remain, including preventing representation collapse, balancing task-specific optimization, improving computational efficiency, and ensuring robustness to distribution shifts.

Future research in SSL should focus on addressing these challenges, particularly through: (1) developing more efficient methods that perform well with smaller batch sizes; (2) exploring novel regularization approaches that promote desired representational properties; (3) designing SSL techniques specifically tailored to transformer architectures; (4) establishing more comprehensive evaluation protocols that assess robustness and OOD generalization; and (5) creating benchmarks that better reflect the diversity and imperfections of real-world data.

As the field continues to develop, SSL has the potential to transform how we approach computer vision tasks, reducing reliance on labeled data while improving generalization to diverse real-world scenarios. By learning from vast amounts of unlabeled data, these methods may ultimately capture more nuanced and comprehensive visual representations than is possible through supervised learning alone.

References

[1] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc., ??? (2012). https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (2021)

[3] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention (2021). https://arxiv.org/abs/2012.12877

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need (2023)

[5] Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A.S., Al-dabbagh, B.S.N., Fadhel, M.A., Manoufali, M., Zhang, J., Al-Timemy, A.H., Duan, Y., Abdullah, A., Farhan, L., Lu, Y., Gupta, A., Albu, F., Abbosh, A., Gu, Y.: A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. Journal of Big Data **10**(1) (2023) https://doi.org/10.1186/s40537-023-00727-2

[6] Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives (2014). https://arxiv.org/abs/1206.5538

[7] Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A.G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., Goldblum, M.: A Cookbook of Self-Supervised Learning (2023)

[8] Ohri, K., Kumar, M.: Review on self-supervised image recognition using deep neural networks. Knowledge-Based Systems **224**, 107090 (2021) https://doi.org/10.1016/j.knosys.2021.107090

[9] Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., Brain, G.: Time-contrastive networks: Self-supervised learning from video. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1134–1141 (2018). https://doi.org/10.1109/ICRA.2018.8462891

[10] Noroozi, M., Favaro, P.: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. arXiv (2016). https://doi.org/10.48550/ARXIV.1603.

09246 . https://arxiv.org/abs/1603.09246

[11] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations (2020)

[12] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)

[13] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners (2020)

[14] Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to Answer Open-Domain Questions (2017)

[15] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc., ??? (2017). https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf

[16] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum Contrast for Unsupervised Visual Representation Learning (2020)

[17] Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., Bojanowski, P.: Self-supervised Pretraining of Visual Features in the Wild (2021)

[18] Goyal, P., Duval, Q., Seessel, I., Caron, M., Misra, I., Sagun, L., Joulin, A., Bojanowski, P.: Vision Models Are More Robust And Fair When Pretrained On Uncurated Images Without Supervision (2022)

[19] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248–255 (2009)

[20] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge (2015)

[21] Oord, A., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding (2019)

[22] Zhang, K., Wen, Q., Zhang, C., Cai, R., Jin, M., Liu, Y., Zhang, J., Liang, Y.,

Pang, G., Song, D., Pan, S.: Self-Supervised Learning for Time Series Analysis: Taxonomy, Progress, and Prospects (2024)

[23] Liu, S., Mallol-Ragolta, A., Parada-Cabaleiro, E., Qian, K., Jing, X., Kathan, A., Hu, B., Schuller, B.W.: Audio self-supervised learning: A survey. Patterns **3**(12), 100616 (2022) https://doi.org/10.1016/j.patter.2022.100616

[24] Schiappa, M.C., Rawat, Y.S., Shah, M.: Self-supervised learning for videos: A survey. ACM Computing Surveys **55**(13s), 1–37 (2023) https://doi.org/10.1145/3577925

[25] Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes (2013)

[26] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders, pp. 1096–1103 (2008). https://doi.org/10.1145/1390156.1390294

[27] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks (2014)

[28] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A Simple Framework for Contrastive Learning of Visual Representations (2020)

[29] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised Learning (2020)

[30] Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. CoRR **abs/1505.05192** (2015) 1505.05192

[31] Krishnan, R., Rajpurkar, P., Topol, E.J.: Self-supervised learning in medicine and healthcare. Nature Biomedical Engineering **6**(12), 1346–1352 (2022) https://doi.org/10.1038/s41551-022-00914-1

[32] Wang, Y., Albrecht, C.M., Braham, N.A.A., Mou, L., Zhu, X.X.: Self-supervised learning in remote sensing: A review. IEEE Geoscience and Remote Sensing Magazine **10**(4), 213–247 (2022) https://doi.org/10.1109/MGRS.2022.3198244

[33] Shurrab, S., Duwairi, R.: Self-supervised learning methods and applications in medical imaging analysis: a survey. PeerJ Computer Science **8**, 1045 (2022) https://doi.org/10.7717/peerj-cs.1045

[34] Chowdhury, A., Rosenthal, J., Waring, J., Umeton, R.: Applying self-supervised learning to medicine: Review of the state of the art and medical implementations. Informatics **8**(3), 59 (2021) https://doi.org/10.3390/informatics8030059

[35] Ozbulak, U., Lee, H.J., Boga, B., Anzaku, E.T., Park, H., Messem, A.V., Neve, W.D., Vankerschaver, J.: Know Your Self-supervised Learning: A Survey on Image-based Generative and Discriminative Training (2023)

[36] Rani, V., Nabi, S.T., Kumar, M., Mittal, A., Kumar, K.: Self-supervised learning: A succinct review. Archives of Computational Methods in Engineering **30**(4), 2761–2775 (2023) https://doi.org/10.1007/s11831-023-09884-2

[37] Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A Survey on Contrastive Self-supervised Learning (2020)

[38] Jing, L., Tian, Y.: Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey (2019)

[39] Kumar, P., Rawat, P., Chauhan, S.: Contrastive self-supervised learning: review, progress, challenges and future research directions. International Journal of Multimedia Information Retrieval **11**(4), 461–488 (2022) https://doi.org/10.1007/s13735-022-00245-6

[40] Khan, A., AlBarri, S., Manzoor, M.A.: Contrastive self-supervised learning: A survey on different architectures. In: 2022 2nd International Conference on Artificial Intelligence (ICAI), pp. 1–6 (2022). https://doi.org/10.1109/ICAI55435.2022.9773725

[41] Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J.: Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering, 1–1 (2021) https://doi.org/10.1109/tkde.2021.3090866

[42] Hu, H., Wang, X., Zhang, Y., Chen, Q., Guan, Q.: A comprehensive survey on contrastive learning. Neurocomputing **610**, 128645 (2024) https://doi.org/10.1016/j.neucom.2024.128645

[43] Zhang, R., Isola, P., Efros, A.A.: Colorful Image Colorization (2016)

[44] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. CoRR **abs/1604.07379** (2016) 1604.07379

[45] Noroozi, M., Vinjimoor, A., Favaro, P., Pirsiavash, H.: Boosting self-supervised learning via knowledge transfer. CoRR **abs/1805.00385** (2018) 1805.00385

[46] Gidaris, S., Singh, P., Komodakis, N.: Unsupervised Representation Learning by Predicting Image Rotations (2018)

[47] Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT Pre-Training of Image Transformers (2022)

[48] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2015)

[49] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition (2015)

[50] Zagoruyko, S., Komodakis, N.: Wide Residual Networks (2016)

[51] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows (2021)

[52] El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., Jegou, H.: XCiT: Cross-Covariance Image Transformers (2021)

[53] Chen, X., He, K.: Exploring Simple Siamese Representation Learning (2020)

[54] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. In: Cowan, J., Tesauro, G., Alspector, J. (eds.) Advances in Neural Information Processing Systems, vol. 6. Morgan-Kaufmann, ??? (1993). https://proceedings.neurips.cc/paper_files/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf

[55] Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 539–5461 (2005). https://doi.org/10.1109/CVPR.2005.202

[56] Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research **10**(9), 207–244 (2009)

[57] Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds.) Pattern Recognition and Image Analysis, pp. 11–14. Springer, Berlin, Heidelberg (2009)

[58] Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 29. Curran Associates, Inc., ??? (2016). https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf

[59] Wu, Z., Xiong, Y., Yu, S., Lin, D.: Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination (2018)

[60] Yeh, C.-H., Hong, C.-Y., Hsu, Y.-C., Liu, T.-L., Chen, Y., LeCun, Y.: Decoupled

Contrastive Learning (2022)

[61] Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations (2021)

[62] Mitrovic, J., McWilliams, B., Walker, J., Buesing, L., Blundell, C.: Representation Learning via Invariant Causal Mechanisms (2020)

[63] Li, J., Zhou, P., Xiong, C., Hoi, S.C.H.: Prototypical Contrastive Learning of Unsupervised Representations (2021)

[64] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 9912–9924. Curran Associates, Inc., ??? (2020). https://proceedings.neurips.cc/paper_files/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf

[65] Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: Learning Augmentation Policies from Data (2019)

[66] Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: RandAugment: Practical automated data augmentation with a reduced search space (2019)

[67] Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep Clustering for Unsupervised Learning of Visual Features (2019)

[68] Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow Twins: Self-Supervised Learning via Redundancy Reduction (2021)

[69] Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. CoRR **abs/1603.06668** (2016) 1603.06668

[70] Caron, M., Bojanowski, P., Mairal, J., Joulin, A.: Unsupervised Pre-Training of Image Features on Non-Curated Data (2019)

[71] Zhang, R., Isola, P., Efros, A.A.: Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction (2017)

[72] Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. CoRR **abs/1605.09782** (2016) 1605.09782

[73] Jenni, S., Jin, H., Favaro, P.: Steering Self-Supervised Feature Learning Beyond Local Pixel Statistics (2020)

[74] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot,

A., Liu, C., Krishnan, D.: Supervised Contrastive Learning (2021)

[75] Tian, Y., Krishnan, D., Isola, P.: Contrastive Multiview Coding (2020)

[76] Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 6827–6839. Curran Associates, Inc., ??? (2020). https://proceedings.neurips.cc/paper_files/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf

[77] Misra, I., Maaten, L.: Self-Supervised Learning of Pretext-Invariant Representations (2019)

[78] Ye, M., Zhang, X., Yuen, P.C., Chang, S.-F.: Unsupervised Embedding Learning via Invariant and Spreading Instance Feature (2019)

[79] Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization (2019)

[80] Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning Representations by Maximizing Mutual Information Across Views (2019)

[81] Hénaff, O.J., Srinivas, A., Fauw, J.D., Razavi, A., Doersch, C., Eslami, S.M.A., Oord, A.: Data-Efficient Image Recognition with Contrastive Predictive Coding (2020)

[82] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big Self-Supervised Models are Strong Semi-Supervised Learners (2020)

[83] Chakraborty, S., Gosthipaty, A.R., Paul, S.: G-SimCLR : Self-Supervised Contrastive Learning with Guided Projection via Pseudo Labelling (2020)

[84] Kumar, V., Nandi, G.C., Kala, R.: Static hand gesture recognition using stacked denoising sparse autoencoders. 2014 Seventh International Conference on Contemporary Computing (IC3), 99–104 (2014)

[85] Yang, H., Ding, X., Wang, J., Li, J.: Simcl:simple contrastive learning for image classification. In: Proceedings of the 5th International Conference on Big Data Technologies. ICBDT 2022. ACM, ??? (2022). https://doi.org/10.1145/3565291.3565335 . http://dx.doi.org/10.1145/3565291.3565335

[86] Lee, K.-H., Arnab, A., Guadarrama, S., Canny, J., Fischer, I.: Compressive Visual Representations (2021)

[87] Jing, L., Vincent, P., LeCun, Y., Tian, Y.: Understanding Dimensional Collapse in Contrastive Self-supervised Learning (2022). https://arxiv.org/abs/2110.09348

[88] Chen, X., Fan, H., Girshick, R., He, K.: Improved Baselines with Momentum Contrastive Learning (2020)

[89] Chen, X., Xie, S., He, K.: An Empirical Study of Training Self-Supervised Vision Transformers (2021)

[90] Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (2019)

[91] You, Y., Gitman, I., Ginsburg, B.: Large Batch Training of Convolutional Networks (2017)

[92] Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard Negative Mixing for Contrastive Learning (2020)

[93] Giakoumoglou, N., Stathaki, T.: SynCo: Synthetic Hard Negatives for Contrastive Visual Representation Learning (2024). https://arxiv.org/abs/2410.02401

[94] Zhang, C., Zhang, K., Pham, T.X., Niu, A., Qiao, Z., Yoo, C.D., Kweon, I.S.: Dual Temperature Helps Contrastive Learning Without Many Negative Samples: Towards Understanding and Simplifying MoCo (2022)

[95] Xiao, T., Wang, X., Efros, A.A., Darrell, T.: What Should Not Be Contrastive in Contrastive Learning (2021)

[96] Zhu, J., Moraes, R.M., Karakulak, S., Sobol, V., Canziani, A., LeCun, Y.: TiCo: Transformation Invariance and Covariance Contrast for Self-Supervised Visual Representation Learning (2022)

[97] Zheng, M., You, S., Wang, F., Qian, C., Zhang, C., Wang, X., Xu, C.: ReSSL: Relational Self-Supervised Learning with Weak Augmentation (2021)

[98] Xie, Z., Lin, Y., Yao, Z., Zhang, Z., Dai, Q., Cao, Y., Hu, H.: Self-Supervised Learning with Swin Transformers (2021)

[99] Tian, Y., Henaff, O.J., Oord, A.: Divide and Contrast: Self-supervised Learning from Uncurated Data (2021)

[100] Koohpayegani, S.A., Tejankar, A., Pirsiavash, H.: Mean Shift for Self-Supervised Learning (2021)

[101] Estepa, I.G., Sarasúa, I., Nagarajan, B., Radeva, P.: All4One: Symbiotic Neighbour Contrastive Learning via Self-Attention and Redundancy Reduction

(2023)

[102] Azabou, M., Azar, M.G., Liu, R., Lin, C.-H., Johnson, E.C., Bhaskaran-Nair, K., Dabagia, M., Avila-Pires, B., Kitchell, L., Hengen, K.B., Gray-Roncal, W., Valko, M., Dyer, E.L.: Mine Your Own vieW: Self-Supervised Learning Through Across-Sample Prediction (2021)

[103] Zhou, P., Zhou, Y., Si, C., Yu, W., Ng, T.K., Yan, S.: Mugs: A Multi-Granular Self-Supervised Learning Framework (2022)

[104] Falcon, W., Cho, K.: A Framework For Contrastive Self-Supervised Learning And Designing A New Approach (2020)

[105] Tomasev, N., Bica, I., McWilliams, B., Buesing, L., Pascanu, R., Blundell, C., Mitrovic, J.: Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet? (2022)

[106] Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., Dai, J.: Exploring the Equivalence of Siamese Self-Supervised Learning via A Unified Gradient Framework (2022)

[107] Ho, C.-H., Vasconcelos, N.: Contrastive Learning with Adversarial Examples (2020). https://arxiv.org/abs/2010.12050

[108] Kim, M., Tack, J., Hwang, S.J.: Adversarial Self-Supervised Contrastive Learning (2020). https://arxiv.org/abs/2006.07589

[109] Chen, K., Zhou, H., Chen, Y., Mao, X., Li, Y., He, Y., Xue, H., Zhang, W., Yu, N.: Self-supervised Adversarial Training (2020). https://arxiv.org/abs/1911.06470

[110] Hu, Q., Wang, X., Hu, W., Qi, G.-J.: Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1074–1083 (2021)

[111] Wang, X., Huang, Y., Zeng, D., Qi, G.-J.: Caco: Both positive and negative samples are directly learnable via cooperative-adversarial contrastive learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[112] Zhang, C., Zhang, K., Zhang, C., Niu, A., Feng, J., Yoo, C.D., Kweon, I.S.: Decoupled Adversarial Contrastive Learning for Self-supervised Adversarial Robustness (2022). https://arxiv.org/abs/2207.10899

[113] Haghighi, F., Taher, M.R.H., Gotway, M.B., Liang, J.: DiRA: Discriminative, Restorative, and Adversarial Learning for Self-supervised Medical Image Analysis (2022). https://arxiv.org/abs/2204.10437

[114] Shah, A., Sra, S., Chellappa, R., Cherian, A.: Max-Margin Contrastive Learning (2021)

[115] Yerxa, T., Kuang, Y., Simoncelli, E., Chung, S.: Learning Efficient Coding of Natural Images with Maximum Manifold Capacity Representations (2023)

[116] Rojas-Gomez, R.A., Singhal, K., Etemad, A., Bijamov, A., Morningstar, W.R., Mansfield, P.A.: SASSL: Enhancing Self-Supervised Learning via Neural Style Transfer (2024). https://arxiv.org/abs/2312.01187

[117] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (2016)

[118] Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense Contrastive Learning for Self-Supervised Visual Pre-Training (2021)

[119] Pinheiro, P.O., Almahairi, A., Benmalek, R.Y., Golemo, F., Courville, A.: Unsupervised Learning of Dense Visual Representations (2020)

[120] Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate Yourself: Exploring Pixel-Level Consistency for Unsupervised Visual Representation Learning (2021)

[121] Hénaff, O.J., Koppula, S., Alayrac, J.-B., Oord, A., Vinyals, O., Carreira, J.: Efficient Visual Pretraining with Contrastive Detection (2021)

[122] Xiao, T., Reed, C.J., Wang, X., Keutzer, K., Darrell, T.: Region Similarity Representation Learning (2021)

[123] Yang, C., Wu, Z., Zhou, B., Lin, S.: Instance Localization for Self-supervised Detection Pretraining (2021)

[124] Girshick, R.: Fast R-CNN (2015)

[125] Buettner, K., Kovashka, A.: Contrastive View Design Strategies to Enhance Robustness to Domain Shifts in Downstream Object Detection (2022)

[126] Bardes, A., Ponce, J., LeCun, Y.: VICRegL: Self-Supervised Learning of Local Visual Features (2022)

[127] Wang, F., Wang, H., Wei, C., Yuille, A., Shen, W.: CP2: Copy-Paste Contrastive Pretraining for Semantic Segmentation (2022)

[128] Wei, F., Gao, Y., Wu, Z., Hu, H., Lin, S.: Aligning Pretraining for Detection via Object-Level Contrastive Learning (2021). https://arxiv.org/abs/2106.02637

[129] Xie, E., Ding, J., Wang, W., Zhan, X., Xu, H., Sun, P., Li, Z., Luo, P.: DetCo:

Unsupervised Contrastive Learning for Object Detection (2021). https://arxiv.org/abs/2102.04803

[130] Yang, J., Zhang, K., Cui, Z., Su, J., Luo, J., Wei, X.: InsCon:Instance Consistency Feature Representation via Self-Supervised Learning (2022). https://arxiv.org/abs/2203.07688

[131] Li, Z., Zhu, Y., Yang, F., Li, W., Zhao, C., Chen, Y., Chen, Z., Xie, J., Wu, L., Zhao, R., Tang, M., Wang, J.: UniVIP: A Unified Framework for Self-Supervised Visual Pre-training (2022)

[132] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context (2015)

[133] Zhuang, C., Zhai, A.L., Yamins, D.: Local Aggregation for Unsupervised Learning of Visual Embeddings (2019)

[134] Zhan, X., Xie, J., Liu, Z., Ong, Y.S., Loy, C.C.: Online Deep Clustering for Unsupervised Representation Learning (2020)

[135] YM., A., C., R., A., V.: Self-labelling via simultaneous clustering and representation learning. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=Hyx-jyBFPr

[136] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Burges, C.J., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26. Curran Associates, Inc., ??? (2013). https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf

[137] Qian, Q., Xu, Y., Hu, J., Li, H., Jin, R.: Unsupervised Visual Representation Learning by Online Constrained K-Means (2022)

[138] Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing Network Design Spaces (2020)

[139] Gansbeke, W.V., Vandenhende, S., Georgoulis, S., Proesmans, M., Gool, L.V.: SCAN: Learning to Classify Images without Labels (2020)

[140] Pang, B., Zhang, Y., Li, Y., Cai, J., Lu, C.: Unsupervised Visual Representation Learning by Synchronous Momentum Grouping (2022)

[141] Amrani, E., Karlinsky, L., Bronstein, A.: Self-Supervised Classification Network (2022)

[142] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin,

A.: Emerging Properties in Self-Supervised Vision Transformers (2021)

[143] Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning Robust Visual Features without Supervision (2023)

[144] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: iBOT: Image BERT Pre-Training with Online Tokenizer (2022)

[145] Sablayrolles, A., Douze, M., Schmid, C., Jégou, H.: Spreading vectors for similarity search (2019)

[146] Pototzky, D., Sultan, A., Schmidt-Thieme, L.: FastSiam: Resource-Efficient Self-supervised Learning on a Single GPU, pp. 53–67. Springer, ??? (2022). https://doi.org/10.1007/978-3-031-16788-1_4 . http://dx.doi.org/10.1007/978-3-031-16788-1_4

[147] Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., Pérez, P.: Learning representations by predicting bags of visual words. In: CVPR (2021)

[148] Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network (2015)

[149] Tian, Y., Krishnan, D., Isola, P.: Contrastive Representation Distillation (2022)

[150] Giakoumoglou, N., Stathaki, T.: Relational Representation Distillation (2024). https://arxiv.org/abs/2407.12073

[151] Giakoumoglou, N., Stathaki, T.: Discriminative and Consistent Representation Distillation (2024). https://arxiv.org/abs/2407.11802

[152] Song, K., Xie, J., Zhang, S., Luo, Z.: Multi-Mode Online Knowledge Distillation for Self-Supervised Visual Representation Learning (2023). https://arxiv.org/abs/2304.06461

[153] Xu, G., Liu, Z., Li, X., Loy, C.C.: Knowledge Distillation Meets Self-Supervision (2020). https://arxiv.org/abs/2006.07114

[154] Dosovitskiy, A., Fischer, P., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks (2015). https://arxiv.org/abs/1406.6909

[155] Bhat, P., Arani, E., Zonooz, B.: Distill on the Go: Online knowledge distillation in self-supervised learning (2021). https://arxiv.org/abs/2104.09866

[156] Fang, Z., Wang, J., Wang, L., Zhang, L., Yang, Y., Liu, Z.: SEED: Self-supervised Distillation For Visual Representation (2021)

[157] Gao, Y., Zhuang, J.-X., Lin, S., Cheng, H., Sun, X., Li, K., Shen, C.: DisCo: Remedy Self-supervised Learning on Lightweight Models with Distilled Contrastive Learning (2022)

[158] Xu, H., Fang, J., Zhang, X., Xie, L., Wang, X., Dai, W., Xiong, H., Tian, Q.: Bag of Instances Aggregation Boosts Self-supervised Distillation (2022)

[159] Hotelling, H.: In: Kotz, S., Johnson, N.L. (eds.) Relations Between Two Sets of Variates, pp. 162–190. Springer, New York, NY (1992). https://doi.org/10.1007/978-1-4612-4380-9_14 . https://doi.org/10.1007/978-1-4612-4380-9_14

[160] Bardes, A., Ponce, J., LeCun, Y.: VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning (2022)

[161] Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for Self-Supervised Representation Learning (2021)

[162] Bandara, W.G.C., Melo, C.M.D., Patel, V.M.: Guarding Barlow Twins Against Overfitting with Mixed Samples (2023)

[163] Zhang, S., Qiu, L., Zhu, F., Yan, J., Zhang, H., Zhao, R., Li, H., Yang, X.: Align representations with base: A new approach to self-supervised learning. In: The IEEE / CVF Computer Vision and Pattern Recognition Conference (2022)

[164] Strang, G.: Linear Algebra and Its Applications. Thomson, Brooks/Cole, Belmont, CA (2006). http://www.amazon.com/Linear-Algebra-Its-Applications-Edition/dp/0030105676

[165] Kalantidis, Y., Lassance, C., Almazan, J., Larlus, D.: TLDR: Twin Learning for Dimensionality Reduction (2022)

[166] Wang, F., Kong, T., Zhang, R., Liu, H., Li, H.: Self-Supervised Learning by Estimating Twin Class Distributions (2021)

[167] Wang, G., Wang, K., Wang, G., Torr, P.H.S., Lin, L.: Solving Inefficiency of Self-supervised Representation Learning (2021)

[168] He, J., Du, J., Ma, W.: Preventing Dimensional Collapse in Self-Supervised Learning via Orthogonality Regularization (2024). https://arxiv.org/abs/2411.00392

[169] Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting Self-Supervised Visual Representation Learning (2019)

[170] Ruder, S.: An overview of gradient descent optimization algorithms (2017)

[171] Kornblith, S., Shlens, J., Le, Q.V.: Do Better ImageNet Models Transfer Better? (2019)

[172] Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., Houlsby, N.: A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark (2020)

[173] Krizhevsky, A.: Learning multiple layers of features from tiny images, 32–33 (2009)

[174] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(6), 1452–1464 (2018) https://doi.org/10.1109/TPAMI.2017.2723009

[175] Yalniz, I.Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification (2019)

[176] iNaturalist. https://www.inaturalist.org. Accessed: 20 November 2024

[177] Wang, X., Qi, G.-J.: Contrastive Learning with Stronger Augmentations (2022)

[178] Wang, Y., Chen, H., Fan, Y., Sun, W., Tao, R., Hou, W., Wang, R., Yang, L., Zhou, Z., Guo, L.-Z., Qi, H., Wu, Z., Li, Y.-F., Nakamura, S., Ye, W., Savvides, M., Raj, B., Shinozaki, T., Schiele, B., Wang, J., Xie, X., Zhang, Y.: USB: A Unified Semi-supervised Learning Benchmark for Classification (2022). https://arxiv.org/abs/2208.07204

[179] Bossard, L., Guillaumin, M., Gool, L.V.: Food-101 – mining discriminative components with random forests. In: Computer Vision – ECCV 2014, pp. 446–461. Springer, ??? (2014). https://doi.org/10.1007/978-3-319-10599-4_29 . https://doi.org/10.1007/978-3-319-10599-4_29

[180] Berg, T., Liu, J., Lee, S.W., Alexander, M.L., Jacobs, D.W., Belhumeur, P.N.: Birdsnap: Large-scale fine-grained visual categorization of birds. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2019–2026 (2014). https://doi.org/10.1109/CVPR.2014.259

[181] Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492 (2010). https://doi.org/10.1109/CVPR.2010.5539970

[182] Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 2013 IEEE International Conference on Computer Vision Workshops, pp. 554–561 (2013). https://doi.org/10.1109/ICCVW.2013.77

[183] Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-Grained Visual Classification of Aircraft (2013)

[184] Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. International Journal of Computer Vision **88**(2), 303–338 (2009) https://doi.org/10.1007/s11263-009-0275-4

[185] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3606–3613 (2014). https://doi.org/10.1109/CVPR.2014.461

[186] Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3498–3505 (2012). https://doi.org/10.1109/CVPR.2012.6248092

[187] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding **106**(1), 59–70 (2007) https://doi.org/10.1016/j.cviu.2005.09.012 . Special issue on Generative Model Based Vision

[188] Nilsback, M.-E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing, pp. 722–729 (2008). https://doi.org/10.1109/ICVGIP.2008.47

[189] Chen, L.-C., Papandreou, G., Schroff, F., Adam, H.: Rethinking Atrous Convolution for Semantic Image Segmentation (2017)

[190] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)

[191] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN (2018)

[192] Gupta, A., Dollár, P., Girshick, R.: LVIS: A Dataset for Large Vocabulary Instance Segmentation (2019)

[193] Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic Understanding of Scenes through the ADE20K Dataset (2018)

[194] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban

Scene Understanding (2016)

[195] Nathan Silberman, P.K. Derek Hoiem, Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)

[196] Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation (2015)

[197] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper Depth Prediction with Fully Convolutional Residual Networks (2016)

[198] Barrett, N., Sadeghi, Z., Matwin, S.: Evolutionary augmentation policy optimization for self-supervised learning. Advances in Artificial Intelligence and Machine Learning **03**(02), 1135–1164 (2023) https://doi.org/10.54364/aaiml.2023.1167

[199] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners (2021)

[200] Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: SimMIM: A Simple Framework for Masked Image Modeling (2022)

Table 10: Comparison of architectural and training characteristics across the MoCo family of self-supervised learning methods.

	MoCo	MoCo-v2	MoCo-v3
Backbone	ResNet	ResNet	ViT and ResNet
Negative sampling	Queue/Memory	Queue/Memory	Batch
Loss symmetry	Unidirectional	Unidirectional	Bidirectional symmetrized
Projection head	Linear layer	MLP (no prediction head)	MLP + prediction head (query only)
Optimization	SGD	SGD	AdamW/LARS
Stability mechanisms	Momentum encoder	Momentum encoder	Frozen patch projection, long warmup
Data augmentation	Basic	Enhanced (SimCLR)	Enhanced (SimCLR)
Batch size	Small (256)	Small (256)	Large (4096)
Training duration	200 epochs	200/800 epochs	Up to 1000 epochs

Table 11: Comparison of various contrastive learning methods based on their projection heads, predictors, momentum encoders, memory usage, type of negatives, views, loss functions, and network architectures.

Method	Proj.	Pred.	Mom.	Mem.	Neg.	Views	Loss	Networks
InstDis [59]	✓			✓	mem.	N/A	non-param. softmax	single
PIRL [77]	✓			✓	mem.	Jigsaw	InfoNCE	siamese
UEL [78]	✓				N/A	N/A	neg. log likelihood	siamese
CPC [21, 81]		AR		✓	mem.	image	InfoNCE	single
CMC [75]	✓			✓	mem.	channels	InfoNCE	siamese
SimCLR [28]	✓				batch	aug.	NT-Xent	siamese
SimCLR-v2 [82]	✓			✓	mem.	aug.	NT-Xent	siamese
G-SimCLR [83]	✓				batch	aug.	NT-Xent	siamese
SupCon [74]	✓				N/A	aug.	SupCon	single
SimCL [85]	✓				batch	noise	InfoNCE	single
MoCo [16]	✓		✓	✓	mem.	aug.	InfoNCE	siamese
MoCo-v2 [88]	✓		✓	✓	mem.	aug.	InfoNCE	siamese
MoCo-v3 [89]	✓		✓		batch	aug.	InfoNCE	siamese
MoCHI [92]	✓		✓	✓	mem.	aug.	InfoNCE	siamese
SimMoCo [94]	✓		✓		batch	aug.	DT	siamese
SimCo [94]	✓				batch	aug.	DT	siamese
MOBY [98]	✓	✓	✓	✓	mem.	aug.	InfoNCE	siamese
LooC [95]	✓		✓	✓	mem.	aug.	InfoNCE var.	siamese
MoCLR [99]	✓		✓		batch	aug.	InfoNCE	siamese
TiCo [96]	✓		✓	N/A	N/A	aug.	TiCo	siamese
InfoMin [76]	✓		✓	✓	mem.	aug.	InfoNCE	siamese
ReSSL [97]	✓		✓	✓	mem.	aug.	softmax+CE	siamese
NNCLR [61]	✓	✓	✓	✓	mem.+batch	aug.	Neighbor InfoNCE	siamese
MSF [100]	✓	✓	✓	✓	mem.+N/A	aug.	MSE	siamese
All4One [101]	✓	✓	✓	✓	mem.+batch	aug.	NNCLR+centroid+BT	siamese
MYOW [102]	✓	✓	✓	✓	mem.	aug.	NCS	siamese
Mugs [103]	✓	✓	✓	✓	mem.+cent.	aug.	InfoNCE+CE	siamese
MMCL [114]	✓				batch	aug.	NT-Xent	siamese
MMCR [115]	✓				batch	aug.	nuclear norm	siamese
ReLIC [62]	✓		✓		batch	aug.	InfoNCE+KL div.	siamese
DCL [60]	✓				mem./batch	aug.	DCL/DCLW	siamese
UniGrad [106]	✓		✓		N/A	aug.	UniGrad	siamese
CLSA [177]	✓		✓	✓	mem.	aug.	contrastive+DDL	siamese
AdCo [110]	✓		✓	✓	mem.	aug.	InfoNCE	siamese
CaCo [111]	✓		✓	✓	mem.	aug.	InfoNCE	siamese

Table 12: Comparison of various dense contrastive methods based on their projection heads, predictors, momentum encoders, memory usage, negatives, views, loss functions, and network architectures. RA: RoI Align; c-p: copy-paste; PPM: Pixel Propagation Module.

Method	Proj.	Pred.	Mom.	Mem.	Negatives	Views	Loss	Networks
DenseCL [118]	✓		✓	✓	memory	aug.	InfoNCE+dense	siamese
DetConS [121]	✓				batch	aug.	InfoNCE	siamese
DetConB [121]	✓	✓	✓		batch	aug.	InfoNCE	siamese
InsLoc [123]	✓	RA	✓	✓	memory	c-p	InfoNCE	siamese
CP2 [127]	✓		✓	✓	memory	c-p	InfoNCE	siamese
PixContrast [120]	✓		✓		image	aug.	InfoNCE var.	siamese
PixPro [120]	✓	PPM	✓		views	aug.	InfoNCE var.+cos.	siamese
UniVIP [131]	✓	✓	✓		N/A	aug.	NCS+affinity+scene	siamese

Table 13: Comparison of various clustering methods based on their projection heads, predictors, momentum encoders, memory usage, online updates, prototypes, and loss functions.

Method	Proj.	Pred.	Mom.	Mem.	Online	Proto	SK	Loss
Deep Cluster [67]	✓							logistic
Deeper Cluster [70]	✓							logistic
LA [133]				✓				log-lik.
SeLa [135]	✓	softmax					✓	CE
PCL [63]	✓					✓		InfoNCE+ProtoNCE
SwAV [64]	✓	softmax			✓	✓	✓	CE
ODC [134]	✓			✓	✓	✓		
SMoG [140]	✓	✓	✓		✓	✓		InfoNCE+var.
Self-Classifier [141]	✓				✓			CE var.

Table 14: Comparison of various self-distillation methods based on their projection heads, predictors, momentum encoders, memory usage, stop gradient operations, loss functions, symmetry, and network architectures.

Method	Proj.	Pred.	Mom.	Mem.	stop-grad	Loss	Sym.	Networks
BYOL [29]	✓	✓	✓		✓	MSE	✓	online/target
DINO [142]	✓		✓		✓	softmax+CE	✓	student/teacher
SimSiam [53]	✓	✓			✓	NCS	✓	siamese
oBoW [147]	✓		✓		✓	CE		student/teacher
DINO-v2 [143]	✓		✓		✓	softmax+CE	✓	student/teacher
FastSiam [146]	✓	✓			✓	NCS		siamese

Table 15: Comparison of various knowledge distillation methods based on their projection heads, predictors, momentum encoders, memory usage, stop gradient operations, loss functions, symmetry, and network architectures.

Method	Proj.	Pred.	Mom.	Mem.	stop-grad	Loss	Sym.	Networks
SEED [156]	✓			✓		softmax+CE		student/teacher
DisCo [157]	✓		✓			InfoNCE+MSE		student/mean/teacher
BINGO [158]	✓			✓	✓	inter+intra InfoNCE		student/teacher

Table 16: Comparison of various feature decorrelation methods based on their projection heads, predictors, momentum encoders, memory usage, loss functions, and network architectures.

Method	Proj.	Pred.	Mom.	Mem.	Loss	Networks
Barlow Twins [68]	✓				Cross Correlation	siamese
VICReg [160]	✓				Variance-Invariance-Covariance	siamese
W-MSE [161]	✓				MSE	siamese
Mixed Barlow Twins [162]	✓				Cross Correlation	siamese
TWIST [166]	✓	softmax			Mutual Information	siamese
TLDR [165]	✓				Cross Correlation	siamese
ARB [163]	✓				MSE	siamese
Truncated Triplet [167]	✓		✓		Truncated Triplet	siamese