Semi-Supervised and Unsupervised Deep Visual Learning: A Survey

Yanbei Chen, Massimiliano Mancini, Xiatian Zhu, and Zeynep Akata,

Abstract—State-of-the-art deep learning models are often built at the cost of collecting a large amount of labeled training data. However, the requirement of exhaustive manual annotations may degrade the model's generalizability when learning in the limited-label regime. Semi-supervised learning and unsupervised learning offer promising paradigms to learn from an abundance of unlabeled visual data. Recent progress in these paradigms has indicated the strong benefits of leveraging unlabeled data to improve model generalization and provide better model initialization. In this survey, we review the recent advanced deep learning algorithms on semi-supervised learning (SSL) and unsupervised learning (UL) for visual recognition from a unified perspective. To offer a holistic understanding of the state-of-the-art in these areas, we propose a unified taxonomy. We categorize existing representative SSL and UL algorithms with comprehensive and insightful analysis to highlight their design rationales in different learning scenarios and applications in different computer vision tasks. Lastly, we discuss the emerging trends and open challenges in SSL and UL to shed light on future critical research directions.

Index Terms—Semi-Supervised, Unsupervised, Self-Supervised, Visual Representation Learning, Survey

1 Introduction

Over the last decade, deep learning algorithms and architectures [1], [2] have been constantly pushing the state of the art in a wide variety of computer vision tasks, ranging from object recognition [3], retrieval [4], detection [5], to segmentation [6]. To achieve human-level performance, deep learning models are typically built by supervised training upon a tremendous amount of labeled training data. However, collecting large-scale labeled training sets manually is not only expensive and time-consuming, but may also be legally prohibited due to privacy, security, and ethics restrictions. Moreover, supervised deep learning models tend to memorize the labeled data and incorporate the annotator's bias, which weakens their generalization to new scenarios with unseen data distributions in practice.

With cheaper imaging technologies and easier access to web data, obtaining large unlabeled visual data is no longer challenging. Learning from unlabeled data thus becomes a natural and promising recipe to scale model generalization towards practical scenarios where it is infeasible to collect large labeled training sets that cover all types of visual variations in illumination, viewpoint, resolution, occlusion, and background clutter induced by different scenes, camera positions, times of the day, and weather conditions. Semi-supervised learning [7], [8] and unsupervised learning [9], [10], [11], [12] stand out as two most representative paradigms for leveraging unlabeled data. Built upon different assumptions, these paradigms are often developed independently, but share the same aim to learn better representations and models using unlabeled data.

- *This work was done when Y.Chen was with the University of Tübingen. E-mail: yanbei.chen@gmail.com*
- *M. Mancini is with the University of Tübingen. E-mail: massimiliano.mancini@uni-tuebingen.de*
- *X. Zhu is with the University of Surrey. E-mail: xiatian.zhu@surrey.ac.uk*
- *Z. Akata is with the University of Tübingen, MPI for Informatics and MPI for Intelligent Systems. E-mail: zeynep.akata@uni-tuebingen.de*

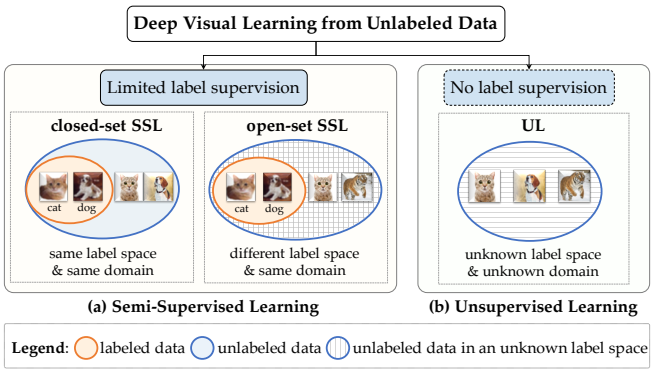


Fig. 1: An overview of semi-supervised and unsupervised learning paradigms – both aim to learn from unlabeled data.

Figure 1 summarizes the two paradigms covered in this survey, which both utilize unlabeled data for visual representation learning. According to whether label annotations are given for a small portion or none of the training data, we categorize the paradigms as semi-supervised learning, and unsupervised learning as defined explicitly in the following.

(a) **Semi-Supervised Learning (SSL)** aims to jointly learn from sparsely labeled data and a large amount of auxiliary unlabeled data often drawn from the same underlying data distribution as the labeled data. In standard closed-set SSL [8], [13], the labeled and unlabeled data belong to the same set of classes from the same domain. In open-set SSL [14], [15], they may not lie in the same label space, i.e., the unlabeled data may contain unknown and/or mislabeled classes.

(b) **Unsupervised Learning (UL)** aims to learn from only unlabeled data without utilizing any task-relevant label supervision. Once trained, the model can be fine-tuned using labeled data to achieve better model generalization in a downstream task [16].

SSL & UL learning objective. Following the above definitions, let the sets of labeled data and unlabeled data be denoted as $\mathcal{D}_l$ and $\mathcal{D}_u$. The overall learning objective for SSL and UL can be written in a unified formulation:

$$\min_{\theta} \lambda_l \sum_{(\mathbf{x},y)\in\mathcal{D}_L} \mathcal{L}_{\text{sup}}(\mathbf{x},y,\theta) + \lambda_u \sum_{\mathbf{x}\in\mathcal{D}_U} \mathcal{L}_{\text{unsup}}(\mathbf{x},\theta), \quad (1)$$

where θ refers to the model parameters of a deep neural network (DNN); $\mathbf{x}$ is an input image and y is the corresponding label; $\mathcal{L}_{\text{sup}}$ and $\mathcal{L}_{\text{unsup}}$ are the supervised and unsupervised loss terms; λ_l and λ_u are balancing hyperparameters. In SSL, both loss terms are jointly optimized. In UL, only the unsupervised loss term is used for unsupervised model pre-training (i.e., $\lambda_l = 0$). Although SSL and UL share the same rationale of learning with an unsupervised objective, they differ in the learning setups, leading to different unique challenges. Specifically, SSL assumes the availability of limited labeled data, and its core challenge is to expand the labeled set with abundant unlabeled data. UL assumes no labeled data for the main learning task and its key challenge is to learn task-generic representations from unlabeled data.

To facilitate the algorithmic understanding of SSL and UL, we focus on providing a timely and comprehensive review of the recent advances in leveraging unlabeled data to improve model generalization, covering the representative state-of-the-art methods in SSL and UL, their application domains, to the emerging trends in self-supervised learning. Importantly, we propose a unified taxonomy of the recent advanced deep learning methods to offer researchers a systematic overview that helps to understand the current state of the art and identify open challenges for future research.

Comparison with previous surveys. Our survey is related to other surveys on semi-supervised learning [8], [13], [17], self-supervised learning [18], [19], or both topics [20]. While these surveys mostly focus on a single particular learning setup [8], [13], [17], [18], non-deep learning methods [8], [13], or lacking a comprehensive taxonomy on methods and discussion on applications [20], our work covers a wider review of representative SSL and UL algorithms involving unlabeled visual data. Importantly, we categorize the state-of-the-art SSL and UL algorithms with novel taxonomies and draw connections among different methods. Beyond intrinsic challenges with each learning paradigm, we distill their underlying connections from the problem and algorithmic perspectives, discuss unique insights into different existing techniques, and their practical applicability.

Survey organization and contributions. Our contributions can be summarized into three aspects. First, to our knowledge, this is the first deep learning survey of its kind to provide a comprehensive review of three prevalent machine learning paradigms in exploiting unlabeled data for visual recognition, including semi-supervised learning (SSL, §2), unsupervised learning (UL, §3), and a further discussion on SSL and UL (§4). Second, we provide a unified, insightful taxonomy and analysis of the existing methods in both the learning setup and model formulation, with an aim at uncovering their underlying algorithmic connections. Finally, we provide an outlook of the recent emerging trends and future research directions in §5 to shed light on those under-explored and potentially critical open avenues.

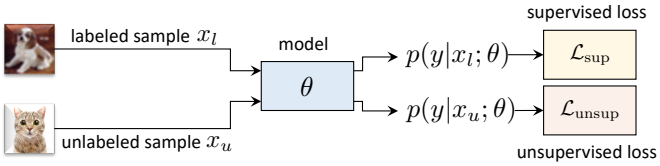


Fig. 2: Semi-supervised learning (SSL) aims to learn jointly from a small set of labeled and a large set of unlabeled data.

2 Semi-Supervised Learning (SSL)

Semi-Supervised Learning (SSL) has been studied in machine learning [8], [13] with an aim at exploiting large unlabeled data together with sparsely labeled data. SSL is explored in various application domains, such as image search [56], medical data analysis [57], web-page classification [58], document retrieval [59], genetics and genomics [60]. More recently, SSL has been used for learning generic visual representations to facilitate many computer vision tasks such as image classification [26], [28], image retrieval [61], object detection [62], [63], semantic segmentation [64], [65], [66], and pose estimation [42], [67], [68]. While our review mainly covers generic semi-supervised learners for image classification [22], [26], [28], [30], the general ideas behind these methods can also be generalized to solve other vision recognition tasks.

We define the SSL problem setup and discuss its assumptions in §2.1. We provide a taxonomy and analysis of the existing semi-supervised deep learning methods in §2.2.

2.1 The Problem Setting of SSL

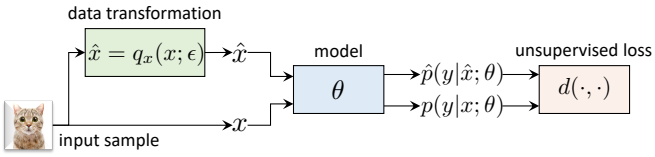
Problem Definition. In SSL, we often have access to a limited amount of labeled samples $\mathcal{D}_l = \{\mathbf{x}_{i,l}, y_i\}_{i=1}^{N_l}$ and a large amount of unlabeled samples $\mathcal{D}_u = \{\mathbf{x}_{i,u}\}_{i=1}^{N_u}$. Each labeled sample $\mathbf{x}_{i,l}$ belongs to one of K class labels $\mathcal{Y} = \{y_k\}_{k=1}^{K}$. For training, the SSL loss function $\mathcal{L}$ for a deep neural network (DNN) θ can generally be expressed as Eq. (1), i.e., $\mathcal{L} = \lambda_l \mathcal{L}_{\text{sup}} + \lambda_u \mathcal{L}_{\text{unsup}}$. In many SSL methods, the hyperparameters λ_u in Eq. (1) is often a ramp-up weighting function (i.e., $\lambda = w(t)$ and t is training iteration), which gradually increases the importance of the unsupervised loss term during training [14], [22], [35], [36], [69]. At test time, the model is deployed to recognize the K known classes. See Figure 2 for an illustration of SSL.

Evaluation Protocol. To test whether an SSL model utilizes the unlabeled data effectively, two evaluation criteria are commonly adopted. First, the model needs to outperform its supervised baseline that learns from merely the labeled data. Second, when increasing the proportion of unlabeled samples in the training set, the improved margins upon the supervised baseline are expected to increase accordingly. Overall, these improved margins indicate the effectiveness and robustness of an SSL method.

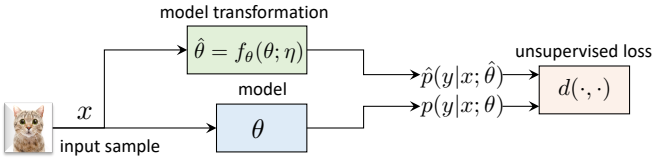
Assumptions. The main assumptions for SSL include the smoothness assumption [70] and manifold assumption [8], [70] – the latter is also known as cluster assumption [71], structure assumption [43], and low-density separation assumption [72]. Specifically, the smoothness assumption considers that the nearby data points are likely to share the same class label. The manifold assumption considers data

TABLE 1: A taxonomy on semi-supervised deep learning methods, including five representative families in §2.2.1 – §2.2.5.

Families of Models	Model Rationale	Representative Strategies and Methods
Consistency regularization	**Random augmentation**	Π-model [21], [22], ensemble transformations [23]
	Adversarial perturbation	Virtual Adversarial Training (VAT) [24], [25]
	MixUp	MixMatch [26], ICT [27]
	Automated augmentation	ReMixMatch [28], UDA [29], FixMatch [30]
	Stochastic perturbation	Pseudo-Ensembles [31], Ladder Network [32], Virtual Adversarial Dropout [33], WCP [34]
	Ensembling	Temporal Ensembling [22], Mean Teacher [35], SWA [36], UASD [14]
Self-training	**Entropy minimization**	Pseudo-Label [37], MixMatch [26], ReMixMatch [28], Memory [38]
	Co-training	Deep Co-training [39], Tri-training [40]
	Distillation	model distillation (Noisy Student Training [41], UASD [14]), data distillation [42]
Graph-based regularization	**Graph-based feature regularizer**	EmbedNN [43], Teacher Graph [44], Graph Convolutional Networks [45]
	Graph-based prediction regularizer	Label Propagation [46]
Deep generative models	**Variational auto-encoders**	Class-conditional VAE [47], ADGM [48]
	Generative adversarial networks	CatGAN [49], FM-GAN [50], ALI [51], BadGAN [52], Localized GAN [53]
Self-supervised learning	**Self-supervision**	S4L [54], SimCLR [12], SimCLRv2 [55]



(a) consistency regularization under input variations



(b) consistency regularization under model variations

Fig. 3: Consistency regularization (§2.2.1) under **(a)** input variations vs **(b)** model variations, where variations can be induced by transformation on input data or model weights.

points lying within the same structure (i.e., the same cluster or manifold) should share the same class label. In other words, the former assumption is imposed locally for nearby data points, while the latter is imposed globally based on the underlying data structure formed by clusters or graphs.

2.2 Taxonomy on SSL Algorithms

Existing SSL methods generally assume that the unlabeled data is closed-set and task-specific, i.e., all unlabeled training samples belong to a pre-defined set of classes. The generic idea shared by most existing works is to assign each unlabeled sample with a class label based on certain underlying data structure, e.g., manifold structure [43], [70], and graph structure [73]. We categorize the most representative semi-supervised deep learning techniques into five categories, including consistency regularization, self-training, graph-based regularization, deep generative model, and self-supervised learning (Figure A). We summarize these categories of methods in Table 1 and detail their general model formulations in §2.2.1, §2.2.2, §2.2.3, §2.2.4 and §2.2.5.

2.2.1 Consistency Regularization

Consistency regularization is a class of powerful techniques that include a number of successful and prevalent SSL approaches [21], [22], [25], [26], [27], [28], [29], [35], [74]. The basic rationale is to enforce consistent model outputs under variations in the input space and (or) model space. The variations are often implemented by adding noises, perturbations or forming variants of the same input or model. Formally, the consistency regularization objective in case of input variation is defined as:

$$\min_{\theta} \sum_{x \in \mathcal{D}} d(p(y|x;\theta), \hat{p}(y|\hat{x};\theta)), \tag{2}$$

and in case of model variation as:

$$\min_{\theta} \sum_{x \in \mathcal{D}} d(p(y|x;\theta), \hat{p}(y|x;\hat{\theta})). \tag{3}$$

In Eq. (2), $\hat{x} = q_x(x;\epsilon)$ is a variant of the original input x, which is derived through a data transformation operation $q_x(\cdot,\epsilon)$ based on certain noise process ϵ, such as data augmentation and stochastic perturbation. Similarly, in Eq. (3), $\hat{\theta} = f_\theta(\theta;\eta)$ is a variant of the model θ derived from a model transformation function $f_\theta(\cdot;\eta)$ based on certain randomness η, such as stochastic perturbation on model weights and model ensembling strategies. In both equations, the consistency is measured as the discrepancy $d(\cdot,\cdot)$ between two network outputs $p(y|\cdot,\cdot)$ and $\hat{p}(y|\cdot,\cdot)$, typically quantified by divergence or distance metrics such as Kullback-Leibler (KL) divergence [25], cross-entropy [29], and mean square error (MSE) [22]. See Figure 3 for an illustration of consistency regularization.

2.2.1.1 Consistency regularization under input variations

To enforce the consistency (i.e., distributional smoothness) under input variations as in Eq. (2), various data transformation strategies can be applied to generate different versions of the same input – denoted as $\hat{x}$ in Eq. (2). A generic visual illustration of consistency regularization under input variations is depicted in Fig. 3 (a). This includes techniques ranging from simple random augmentation [21], [22], to more advanced transformations such as adversarial perturbation [25], MixUp [26], [75], as well as stronger automated augmentation such as AutoAugment [76], RandAugment [77], CTAugment [28] and Cutout [78]. We review these four streams of techniques as follows.

Random augmentation is a standard data transformation strategy widely adopted in consistency regularization [21], [22], [35], including adding Gaussian noise and applying simple domain-specific jittering such as flipping and cropping on image data. One of the representative SSL algorithms, Π-model [21], [22], applies random data augmentation on the same input and minimizes a consistency regularization term (i.e., MSE) between two network outputs. A more recent approach named "ensemble transformations" [23] introduces more diverse data augmentation on input images, including spatial transformations (i.e., projective, affine, similarity, euclidean transformations) to modify the spatial aspect ratio, as well as non-spatial transformations to change the color, contrast, brightness, and sharpness. By enforcing consistency on the ensembled data transformations, the model learns representations invariant to various transformations in semi-supervised and supervised learning.

Adversarial perturbation augments the input data by adding adversarial noise to it. This noise targets at altering the model predictions, e.g., reducing predictive confidence or changing the predicted correct label [79], [80]. Adversarial noise is introduced for SSL to augment data and learn from the unlabeled data with adversarial transformations [24], [25], [74], [81]. Virtual Adversarial Training (VAT) [24], [25] is the first representative SSL method that perturbs input data adversarially. In VAT, a small adversarial perturbation is added to each input and a consistency regularization term (i.e., KL divergence) is imposed to encourage distributional robustness of the model against the virtual adversarial direction. Notably, it has been discovered that semi-supervised learning with adversarial perturbed unlabeled data does not only improve model generalization, but it also enhances robustness to adversarial attacks [81], [82].

MixUp is a simple and data-agnostic augmentation strategy that augments the input data by performing linear interpolations on two inputs and their corresponding labels [75]. It is also introduced as an effective regularizer for SSL [26], [27]. The Interpolation Consistency Training (ICT) [27] interpolates two unlabeled samples and their network outputs. MixMatch [26] further considers to mix a labeled sample and unlabeled sample as the input, and the groundtruth label (of labeled data) and the predicted label (of unlabeled data) as the output targets. Both methods impose consistency regularization to guide the learning of a mapping between the interpolated input and interpolated output for learning from the unlabeled data.

Automated augmentation learns advanced augmentation strategies automatically from data to produce strong samples, alleviating the need for manual design of domain-specific data augmentation [76], [77], [83], [84], [85]. It is also recently introduced for SSL by enforcing the consistency between the prediction of a weakly-augmented or clean sample and the averaged prediction of its strongly augmented versions derived from automated augmentation [28], [29]. Inspired by the recent advances of AutoAugment [76], ReMixMatch [28] introduces CTAugment to learn an automated augmentation policy. Unsupervised Data Augmentation (UDA) [29] adopts RandAugment [77] to produce more diverse and strongly augmented samples by uniformly sampling a set of standard transformations based on the Python Image Library. Later on, FixMatch [30] unifies multiple augmentation strategies including Cutout [78], CTAugment [28], and RandAugment [77] to produce even more strongly augmented samples as input.

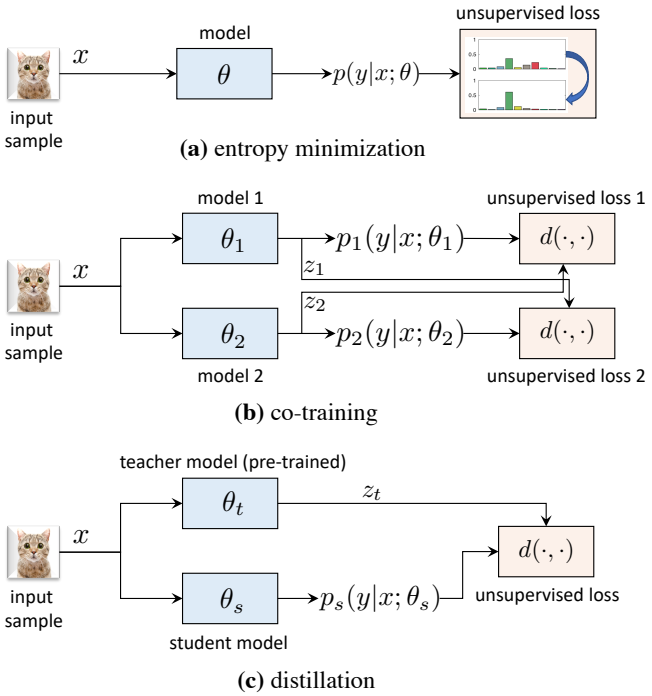
2.2.1.2 *Consistency regularization under model variations*

To impose the predictive consistency under model variations (i.e., variations made in the model's parameter space) as in Eq. (3), stochastic perturbation [31], [32], [33] and ensembling [22], [35], [86] are proposed to generate non-identical models and produce different outputs for the same input – a new model variant is denoted by $\hat{\theta}$ in Eq. (3). A generic visual illustration of consistency regularization under model variations is depicted in Fig. 3 (b). We review two representative streams of works as follows.

Stochastic perturbation introduces slight modifications on the model weights by adding Gaussian noise, dropout, or adversarial noise in a class-agnostic manner [31], [32], [33], [34]. For example, Ladder Network injects layer-wise Gaussian noises into the network and minimizes a denoising L2 loss between outputs from the original network and the noisy-corrupted network [32]. Pseudo-Ensemble applies dropout on the model's parameters to obtain a collection of models (a pseudo-ensemble), while minimizing the disagreements (KL divergence) between the pseudo-ensemble and the model [31]. Similarly, Virtual Adversarial Dropout introduces adversarial dropout to selectively deactivates network neurons and minimizes the discrepancy between outputs from the original model and the perturbed model [33]. A recent approach, Worst-Case Perturbations (WCP) introduces both addictive perturbations and drop connections on model parameters, where drop connections set certain model weights to zero to further change the network structure [34]. Notably, these perturbation mechanisms promote the model robustness against noises in network parameters or structure to learn from unlabeled data (Fig. 3 (b)).

Ensembling is an another effective method to produce a set of models covering different regions of the version space [87], [88], [89]. As demonstrated by seminal machine learning models such as boosting [90] and random forest [89], a set of different models can often provide more reliable predictions than a single model. Moreover, ensembling offers richer inference uncertainty to mitigate the overconfidence issue in deep neural networks [91]. Owing to these merits, ensembling is also introduced in consistency regularization for SSL, where an ensemble model is typically derived by computing an exponential moving average (EMA) or equal average in the prediction space or weight space [14], [22], [35], [36]. Temporal Ensembling [22] and Mean Teacher [35] are two representatives that first propose to ensemble all the networks produced during training by maintaining an EMA in the weight space [35] or prediction space [22]. Stochastic Weight Averaging (SWA) [36] applies an equal average of the model parameters in the weight space to provide a more stable target for deriving the consistency cost. Later on, Uncertainty-Aware Self-Distillation (UASD) [14] computes an equal average of all the preceding model predictions during training to derive soft targets as the regularizer.

Remarks. Consistency regularization can be treated as an auxiliary task that learns from the unlabeled data by mini-



(a) entropy minimization

(b) co-training

(c) distillation

Fig. 4: In self-training, **(a)** the model prediction is enforced to have low entropy, **(b)** two models learn from each other and **(c)** the student model learns from the teacher model.

mizing the model predictive variance towards the variations in the input space or weight space, while the predictive variance is generally quantified as the discrepancy between two predictive probability distributions or network outputs. By minimizing the consistency regularization loss, the model is encouraged to learn more powerful representations that are invariant towards variations added on each sample, without utilizing any additional label annotations.

2.2.2 Self-Training

Self-training is a class of algorithms that learn from the unlabeled data by imputing the labels for samples predicted with high confidence [58], [59], [92]. It is originally proposed for conventional machine learning models such as logistic regression [92], bipartite graph [58] and Naive Bayes classifier [59]. Recently, it is further re-visited in training deep neural networks to learn from massive unlabeled data along with limited labeled data. In the following, we review three representative lines of works in self-training, including entropy minimization, co-training and distillation. See Figure 4 for an illustration of self-training.

Entropy minimization is one of the self-training methods that regularize the model learning based on the low density separation assumption [72], [92], which considers that the class decision boundary should be placed in the low density regions. This is also in line with the cluster assumption and manifold assumption [43], [70], which hypothesizes that data points from the same class are likely to share the same cluster or manifold. Formally, the entropy minimization objective can be formulated as:

$$\min_{\theta} \sum_{x \in \mathcal{D}} \Big(- \sum\nolimits_{j=1}^{K} p(y_j|x;\theta) \log\ p(y_j|x;\theta) \Big), \qquad (4)$$

where K refers to the number of classes. $p(y_j|x;\theta)$ is the probability of assigning the sample x to the class y_j. Eq. (4) is a measure of class overlap. As a lower entropy indicates a higher confidence in model prediction, minimizing Eq. (4) enforces each unlabeled sample to be assigned to the class predicted with the highest probability. Although entropy minimization is originally proposed for logistic regression to impute the labels for samples classified with high confidence [92], it is later extended to train deep neural networks in a semi-supervised fashion by minimizing the entropy of the class assignments that are either derived in the prediction space [25], [26], [28], [30], [37], [93] or the feature space [38], as detailed next.

Entropy minimization can be imposed in the prediction space. For instance, Pseudo-Label [37] directly assigns each sample to the class label predicted with the maximum probability, which implicitly minimizes the entropy of model predictions. When pseudo labels are one-hot, they could easily cause error propagation due to the wrong label assignments. To alleviate this risk, MixMatch [26] uses an ensemble of predictions over different input augmentations, and softly sharpens the one-hot pseudo labels with a temperature hyperparameter. Similarly, FixMatch [30] assigns the one-hot labels only when the confidence scores of the model predictions are higher than a certain threshold.

Entropy minimization can also be imposed in the feature space, as it is also feasible to derive the class assignments based on proximities to class-level prototypes (e.g., class centers) in the feature space [38], [94]. In [38], a Memory module is proposed to learn a feature center per class and the class assignment is derived based on proximities to all the feature centers. By entropy minimization, each unlabeled sample is assigned to the nearest feature center.

Co-training learns two or multiple classifiers on more than one view from different sources that describe the same samples [7], [39], [40], [58], [59]. Conceptually, a co-training framework [58], [59] trains two independent classifier models on two different but complementary data views and imputes the predicted labels in a cross-model manner. It is later extended for deep visual learning [39], [40], [95], as represented by Deep Co-training (DCT) [39] and Tri-training [40]. DCT [39] trains a network with two or more classification layers, and passes different views (e.g., the original view and the adversarial view [96]) to individual classifiers for co-training, while an unsupervised loss is imposed to minimize the similarity of predictions from different views.

The basic idea of co-training can be extended from dual-view [39] to triple [40] or multi-view [39] – e.g., in Tri-training [40], three classifiers are trained together, with labels assigned to the unlabeled data when two of the classifiers agree on the predictions and the confidence scores are higher than a threshold. Formally, the existing deep co-training objective can be written as Eq. (5).

$$\min_{\theta} \sum_{x \in \mathcal{D}} d(p_1(y|x;\theta_1), z_2) + d(p_2(y|x;\theta_2), z_1), \qquad (5)$$

where p_1, p_2 are predictions of two independent classifiers θ_1, θ_2 trained on different data views. $d(\cdot,\cdot)$ introduces the similarity metric to learn from the imputed targets z_1, z_2 from each other, e.g., cross-entropy on one-hot targets [40], or Jensen-Shannon divergence between output targets [39].

Distillation is originally proposed to transfer the knowledge learned by a teacher model to a student model, where the soft targets from the teacher model (e.g., an ensemble of networks or a larger network) can serve as an effective regularizer or a model compression strategy to train a student model [97], [98], [99]. Recent works in SSL use distillation to transfer knowledge learned from the teacher network to impute learning targets on the unlabeled data for training the student network [14], [41], [42], [100]. Formally, an unsupervised distillation objective is introduced on a student model θ_s to learn from the unlabeled data as:

$$\min_{\theta} \sum_{x \in \mathcal{D}} d(p_s(y|x;\theta_s), z_t), \tag{6}$$

where the student prediction p_s is enforced to align with the targets z_t produced by a teacher model θ_t on either the unlabeled data or all the data. Compared to co-training (Eq. (5)), distillation in SSL (Eq. (6)) does not optimize multiple networks simultaneously, but instead trains more than one network in different stages. In distillation, the existing works can be further grouped into model distillation and data distillation, which generate learning targets for unlabeled data using the teacher model output or multiple forward passes of the same input data, as detailed next.

In model distillation, the standard teacher-student learning paradigm is typically adopted to assign labels from a teacher model to a student model [14], [41], [100]. The teacher model can be formed differently, e.g., a pre-trained model or an ensemble of models. In Noisy Student Training [41], an iterative self-training process is proposed to iterate the teacher-student training by first training a teacher to impute labels on unlabeled data for the student, and reusing the student as the teacher in the next iteration. In Uncertainty-Aware Self-Distillation (USAD) [14], the teacher is formed by averaging all the preceding network predictions to impute labels on unlabeled data for updating the student network itself. In model distillation, both soft targets and one-hot labels from the teacher model can serve as the learning targets on the unlabeled data [14], [41].

In data distillation, the teacher model predicts learning targets on unlabeled data by ensembling the outputs of the same input under different data transformations [42]. Specifically, the ensembled teacher predictions (i.e., soft targets) are derived by averaging the outputs of the same inputs under multiple data transformations; while the student model is then trained with the soft targets. Compared to model distillation, data distillation transforms the input data multiple times rather than training multiple networks to impute the ensembled predictions on unlabeled data. This is similar to consistency regularization with random data augmentation; however, in data distillation, two training stages are involved – the first stage involves pre-training the teacher model; while the second stage involves training the student network to mimic the teacher model by distillation.

Remarks. Similar to consistency regularization, self-training can be considered as an unsupervised auxiliary task learned along with the supervised learning task. In general, it also enforces the predictive invariance towards instance-wise variations or the teacher's predictions. However, self-training differs in design. While consistency regularization generally trains one model, self-training may require more than one model to be trained, e.g., co-training requires at least two models co-trained in parallel while distillation requires to train a teacher and a student model sequentially.

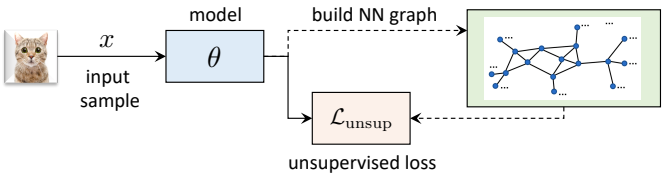


Fig. 5: Graph-based regularization (§2.2.3) for training a neural network with an augmented Nearest Neighbor graph. Pseudo labels are propagated over the graph based on neighbourhood consistency and an unsupervised regularization term is imposed on the feature or prediction space.

2.2.3 Graph-based Regularization

Graph-based regularization is a family of transductive learning methods originally proposed for non-deep semi-supervised learning algorithms [70], [73], [101], [102], [103], such as transductive Support Vector Machine [70], [102] and Gaussian random field model [101]. Most algorithms from this family build a weighted graph to exploit the relationships among the data samples. Specifically, both labeled and unlabeled samples are represented as nodes, while the edge weights encode the similarities between different samples. The labels can be propagated over the graph based on the smoothness assumption [70], i.e., neighboring data points should share the same class label. See Figure 5 for an illustration of graph-based regularization.

To learn from the unlabeled data, a graph-based regularization term is generally imposed for model optimization, which imposes various forms of smoothness constraints to minimize the pairwise similarities between nearby data points. Graph-based regularization is later reformulated for semi-supervised learning with deep neural networks, such as EmbedNN [43], Graph Convolutional Network [45], [104], Teacher Graph [44], and Label Propagation [46]. Although this line of works share the same smoothness assumption for model optimization, graph-based regularization can be imposed differently in either the feature space or prediction space, as detailed in the following.

Graph-based feature regularizer. To impose graph-based regularization, one typical way is to build a learnable nearest neighbor (NN) graph that augments the original DNN to encode the affinity between data points in the feature space, as represented by EmbedCNN [43] and Teacher Graph [44]. Each node in the graph is encoded by the visual feature extracted from the intermediate network layer or the output from the last layer; while an affinity matrix W_{ij} is computed to encode the pairwise similarities between all the nodes. To exploit the unlabeled data, a graph-based regularization term can be formed as a metric learning loss, such as the margin-based contrastive loss for Siamese networks [105], [106] which constrains feature learning by enforcing the local smoothness as follows;

$$\min_{\theta} \sum_{x_i, x_j \in \mathcal{D}} \begin{cases} \|h(x_i) - h(x_j)\|^2, & \text{if } W_{ij}{=}1 \\ \max(0, m - ||h(x_i) - h(x_j)||)^2, & \text{if } W_{ij}{=}0 \end{cases} \tag{7}$$

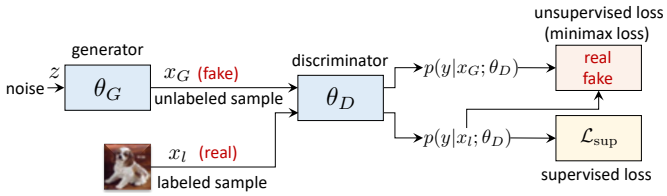


Fig. 6: In GAN-based deep generative models (§2.2.4), the unlabeled data that is generated by an image generator is assigned to an auxiliary class (i.e., K+1) by the discriminator, while the labeled samples are assigned to the K classes. At test time, the discriminator acts as the classifier.

ensuring that features $h(x_i), h(x_j)$ of nearest neighbors (i.e., W_{ij}=1) are close to each other, while pushing the features of dissimilar pairs (i.e., W_{ij}=0) away from each other with a distance margin m.

Beyond augmenting a DNN with a graph, a more flexible way is to use graph convolutions, i.e., Graph Convolutional Networks (GCN) [45], which derive new feature representations for each node subject to the graph structure [104], [107]. Specifically, a GCN takes the data and affinity matrix as input, and learns to estimate the class labels of unlabeled data under a supervised cross-entropy loss on labeled data.

Graph-based prediction regularizer. A graph-based regularization can also be introduced in the prediction space [46], [108], as in Label Propagation [46]. Driven by the same rationale of building a learnable NN-graph as above, in label propagation, an NN-graph is computed to encode the similarity between data points and propagate the labels from the labeled data to the unlabeled data based on transitivity. To train the network, a cross-entropy loss is computed with the graph propagated labels to learn from the unlabeled data. While being similar to the approach Pseudo-Labels [37], the propagated labels are derived with an external NN-graph that encodes the global manifold structure. Further, the label propagation on graph and the update of DNN are performed alternatively to propagate more reliable labels.

Remarks. Graph-based regularization shares several similarities with consistency regularization and self-training in SSL. First, it introduces an unsupervised auxiliary task to train a DNN with propagated learning targets (e.g., pseudo labels) on the unlabeled data. Second, its learning objective can be formulated as a cross-entropy loss or metric learning loss. Notably, while consistency regularization and self-training are inductive approaches that estimate a learning target per instance, graph-based regularization methods are transductive approaches that propagate learning targets based on a graph constructed on the dataset. Beyond concrete details, however, these three techniques all share the same fundamental idea of seeking for unsupervised targets.

2.2.4 Deep Generative Models

Deep generative models are a class of unsupervised learning models that learn to approximate the data distributions without labels [109], [110]. By integrating the generative unsupervised learning concept into a supervised model, a semi-supervised learning framework can be formulated to unify the merits of supervised and unsupervised learning. Two main streams of deep generative models are Variational Auto-Encoders (VAEs) and Generative Adversarial Networks (GANs), as detailed below. See Figure 6 for an illustration of a GAN framework for SSL.

Variational auto-encoders (VAEs) are probabilistic models based on variational inference for unsupervised learning of a complex data distribution [109], [111]. A standard VAE model contains an encoder network that encodes an input sample to a latent variable and a decoder network that decodes the latent variable to reconstruct the input; while a variational lower bound is maximized to train the model. When applied to semi-supervised learning [47], [48], [112], an unsupervised VAE model is generally combined with a supervised classifier. For instance, to predict task-specific class information required in SSL, Class-conditional VAE [47] and ADGM [48] introduce the class label as an extra latent variable in the latent feature space to explicitly disentangle the class information (content) and the stochastic information (style), and impose an explicit classification loss on the labeled data along with the vanilla VAE loss.

Generative adversarial networks (GANs) [110] learn to capture the data distribution by an adversarial minimax game. Specifically, a generator is trained to generate as realistic images as possible while a discriminator is trained to discriminate between real and generated samples. When re-formulated as a semi-supervised representation learner, GANs can leverage the benefits of both unsupervised generative modeling and supervised discriminative learning [49], [50], [51], [52], [53], [113], [114], [115], [116], [117], [118].

The generic idea is to augment the standard GAN framework with supervised learning on the labeled real samples (i.e., discriminative) and unsupervised learning on the generated samples. Formally, this enhances the original discriminator with an extra supervised learning capability. For example, Categorical GAN (CatGAN) [49] introduces a K-class discriminator, and minimizes a supervised cross-entropy loss on the real labeled samples, while imposing a uniform distribution constraint on the generated samples by maximizing the prediction's entropy. Similarly, feature matching GAN (FM-GAN) [50], ALI [51], BadGAN [52] and Localized GAN [53] formulate a $(K+1)$-class discriminator for SSL, whereby a real labeled sample x_l is considered as one of the K classes and a generated sample x_G as the $(K+1)_{\text{th}}$ class. The supervised and unsupervised learning objective for the $(K+1)$-class discriminator can be written as follows;

$$\max_{\theta} \sum_{x \in \mathcal{D}} \log p(y|x_l, y<K+1), \tag{8}$$

$$\max_{\theta} \sum_{x \in \mathcal{D}} \log\,(1-p(y=K+1|x_l)) - \log p(y=K+1|x_G), \tag{9}$$

where Eq. (8) is the supervised classification loss on the labeled samples x_l; Eq. (9) is an unsupervised GAN loss that discriminates between the real labeled samples x_l and the generated fake samples x_G from the image generator. In addition to the GAN-based objectives for model optimization, other objectives could be further introduced to constrain the generated samples. For instance, Localized GAN [53] introduces a regularizer on the generator to ensure the generated samples lie in the neighborhood of an original sample on the manifold, thus allowing to train a locally consistent classifier based on the generated samples in a semi-supervised fashion.

Remarks. Unlike discriminative SSL techniques such as consistency regularization, self-training, and graph-based regularization, DGMs can naturally learn from unlabeled data without the need to estimate their labels. In other words, DGMs are native unsupervised representation learners. To enable SSL in DGMs, the key in model reformulation is thus to integrate the label supervision into training, e.g., adding a class label latent variable in VAEs or an extra class discriminator in GANs. Further, one also needs to tackle more difficult model optimization in a GAN framework.

2.2.5 Self-Supervised Learning

Self-supervised learning is a class of unsupervised representation learners designed based on unsupervised surrogate (pretext) tasks [11], [119], [120], [121], [122], [123]. It is worth noting that self-supervision differs from the self-training algorithms in §2.2.2, as self-supervised learning objectives are task-agnostic and could be trained without any label supervision. Self-supervision is originally proposed to learn from only unlabeled data with task-agnostic unsupervised learning objectives, but recently also be explored for SSL [12], [54], [55]. In semi-supervised training, task-agnostic self-supervision signals on all training data are often integrated with a supervised learning objective on labeled data. For instance, the earlier work S4L [54] uses self-supervision for SSL based on multiple self-supervision signals such as predicting rotation degree [123] and enforcing invariance to exemplar transformation [119] to train the model along with supervised learning. Later on, the follow-up works SimCLR [12] and SimCLRv2 [55] introduce self-supervised contrastive learning for task-agnostic unsupervised pre-training, followed by supervised or semi-supervised fine-tuning using label supervision in the downstream task.

Remarks. A unique advantage of self-supervision for SSL is that task-specific label supervision is not required during training. While the aforementioned semi-supervised learners typically solve a supervised task and an auxiliary unsupervised task jointly, self-supervised semi-supervised learners can be trained in a fully task-agnostic fashion. This suggests the great flexibility of self-supervision for SSL. Thus, the self-supervised training can be introduced as unsupervised pre-training or as an auxiliary unsupervised task solved along with supervised learning. Although self-supervision is relatively new for SSL, it has been more widely explored for unsupervised learning, which is detailed more extensively in §3.2.1 and §3.2.2.

3 Unsupervised Learning (UL)

Unsupervised Learning (UL) aims to learn representations from unlabeled data without utilizing any label supervision. The learned representation is not only expected to capture the underlying semantic information, but also be transferable to tackle unseen downstream tasks such as visual recognition, detection, and segmentation [16], visual retrieval [152], and object tracking [153].

UL is attractive in computer vision for multiple reasons. First, due to the high cost of label annotations, large labeled datasets may not unavailable in many application scenarios, e.g., medical imaging [154]. Second, as there often exists data/label distribution drifts (or gaps) across tasks and application scenarios, pre-training on a large labeled dataset cannot always guarantee good model initialization for unseen situations [155]. Third, UL could supply strong pre-trained models that may perform on par with or even outperform supervised pre-training [12], [16], [156].

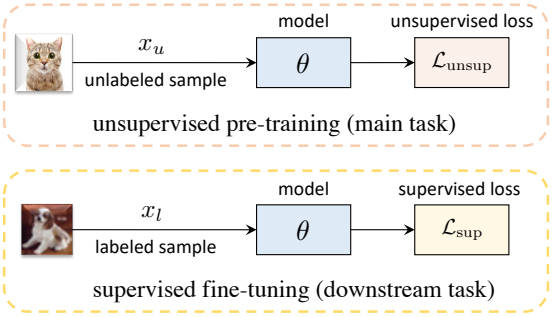


Fig. 7: Unsupervised learning trains a generalizable model using purely unlabeled data. The model can later be fine-tuned with labeled data and tested on a downstream task.

Remarks. **UL** and **SSL** share the same aim to learn from unlabeled data, and leverage similar modeling principles to formulate unsupervised surrogate supervision signals without any label annotation. However, instead of assuming the availability of task-specific information (i.e., class labels) as in SSL, UL considers model learning from purely task-agnostic unlabeled data. Given that unlabeled data are abundantly available in different scenarios (e.g., Internet), UL offers an appealing strategy to provide good pre-trained models that could facilitate various downstream tasks.

This section focuses on generic unsupervised visual learners trained on image classification datasets, which is organized as follows. We define the UL problem setup in §3.1, and provide a taxonomy and analysis of the existing representative unsupervised deep learning methods in §3.2.

3.1 The Problem Setting of UL

Problem Definition. In UL, we have access to an unlabeled dataset $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=1}^{N_u}$. As label information is unknown, the UL loss function $\mathcal{L}$ for training a DNN θ can generally be expressed as Eq. (1), i.e., $\mathcal{L} = \lambda_l \mathcal{L}_{\text{sup}} + \lambda_u \mathcal{L}_{\text{unsup}}$ while $\lambda_l = 0$. In discriminative models, the unsupervised objective $\mathcal{L}_{\text{unsup}}$ requires certain pseudo/proxy targets to learn semantic and generalizable representations. In generative models, $\mathcal{L}_{\text{unsup}}$ is imposed to explicitly model the data distribution. See Figure 7 for an illustration of UL.

Evaluation Protocol. To evaluate the performance of UL methods, two evaluation protocols are often adopted, commonly known as the (1) linear classification protocol, and (2) fine-tuning on downstream tasks. In (1), the pre-trained DNN is frozen to extract the features for an image dataset, while a linear classifier (e.g., a fully-connected layer or a kNN classifier) is trained for classification using the extracted features. In (2), the pre-trained DNN is used to initialize a model for any downstream task, followed by fine-tuning with a task-specific objective, such as fine-tuning an object detector initialized from an unsupervised pre-trained backbone (e.g., FasterR-CNN [157]) on object detection datasets (e.g., PASCAL VOC [158]), or fine-tuning a segmentation model (e.g., Mask R-CNN [159]) with a

TABLE 2: A taxonomy of unsupervised deep learning methods, including three representative families in §3.2.1 – §3.2.3.

Families of Models	Model Rationale	Representative Strategies and Methods	
Pretext tasks	**Pixel-level**	reconstruction [124], [125], inpainting [126], denoising [127], colorization [128], [129], [130]	
	Instance-level	predict image rotations [123], scaling and tiling [122], patch ordering [11], patch re-ordering [121]	
Discriminative models	**Instance discrimination**	negative sampling	large batch size (SimLR [12]), memory bank (InstDis [131]), queue (MoCo [16])
			hard negatives [132], adversarial negatives [133], cooperative positive [134]
		input transformation	data augmentation (PIRL [135]), multi-view augmentation (CMC [136])
		negative-sample-free	simple siamese (SimSiam [137]), Bootstrap (BYOL [138]), DirectPred [139]
	Deep clustering	offline clustering	DeepCluster [140], JULE [141], SeLa [142]
		online clustering	IIC [143], PICA [144], AssociativeCluster [145], SwAV [146]
Deep generative models	**Discriminator-level**	DCGAN [147], Self-supervised GAN [148], Transformation GAN [149]	
	Generator-level	BiGAN [150], BigBiGAN [151]	

pre-trained backbone on segmentation datasets (e.g., COCO [160]). The latter protocol is more critical evaluation.

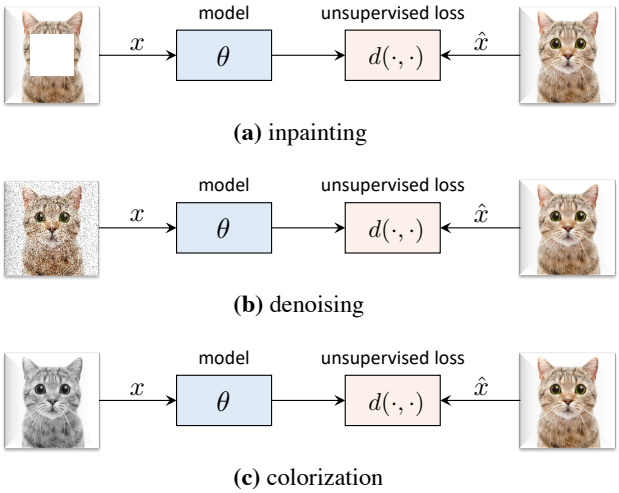
3.2 Taxonomy on UL Algorithms

Existing unsupervised deep learning models can be mainly grouped into three families: pretext tasks, discriminative models and generative models (Figure B). The former two families of UL algorithms are also known as self-supervised learning, which drive model learning by a proxy protocol/task and construct pseudo label supervision to formulate unsupervised surrogate losses. The latter one is inherently unsupervised and explicitly models the data distribution to learn representations without label supervision. We review these three families of models, as summarized in Table 2 and detailed in §3.2.1, §3.2.2 and §3.2.3.

3.2.1 Pretext Tasks

Pretext Tasks refer to hand-crafted proxy tasks manually designed to predict certain task-agnostic properties of the input data, which do not require any label supervision for training. By formulating self-supervised learning objectives with free labels, meaningful visual representations can be learned in a fully unsupervised manner. In the following, we review two classes of pretext tasks, which introduces the self-supervision signals at the pixel-level or instance-level. See Figure 8 and Figure 9 for the illustration of pixel-level pretext tasks and instance-level pretext tasks, which are detailed in the following.

Pixel-level pretext task is generally designed as a dense prediction task that aims to predict the expected pixel values of an output image as a self-supervision signal [124], [125], [126], [127], [128], [129], [130], [161]. Auto-Encoder [124], [125] is one of the most representative and primitive unsupervised models that learn representations by reconstructing input images. In addition to standard reconstruction, recent pixel-level pretext tasks introduce more advanced image generation tasks to hallucinate the pixel colour values of the corrupted input images, as represented by three standard low-level image processing tasks: (1) image inpainting [126], [161] learns by inpainting the masked-out missing regions in the input images; (2) denoising [127] learns to denoise the partial destructed input; and (3) colorization [128], [129], [130] aims to predict the colour values of the grayscale images. These self-supervised models are trained



(a) inpainting

(b) denoising

(c) colorization

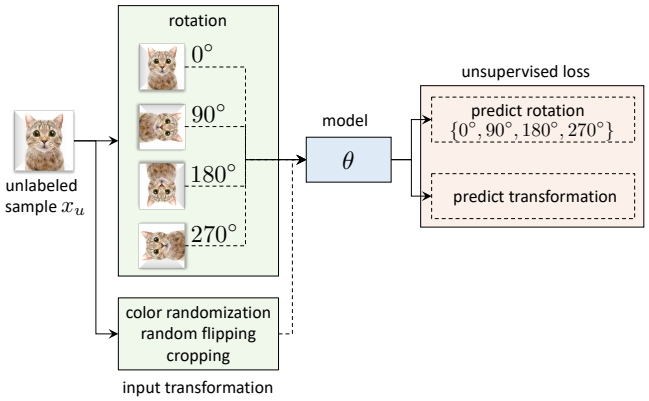
Fig. 8: In pixel-level pretext tasks (§3.2.1), **(a)** inpainting, **(b)** denoising, **(c)** colorization, the model aims to reconstruct the original image ($\hat{x}$) from a corrupted input image (x).

with an image generation task objective (e.g., a mean square error) to enforce predicting the expected pixel values:

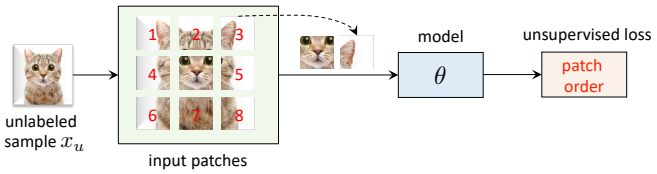
$$\min_{\theta} \sum_{x \in \mathcal{D}} ||G_\theta(x) - \hat{x}||^2, \tag{10}$$

where $G_\theta(\cdot)$ is an image generation network (typically implemented as an encoder-decoder network architecture) trained to predict the expected output image $\hat{x}$ per pixel. Once trained, part of the network $G_\theta(\cdot)$ (e.g., encoder) can be used to initialize the model weights or extract the intermediate features for solving the downstream task.

Instance-level pretext tasks introduce sparse semantic labels for each image sample by designing a surrogate proxy task that can be solved per instance without any label annotations [11], [121], [122], [123], [162], [163], [164], [165]. In general, these pretext tasks involve applying different image transformations to generate diverse input variations, whereby an artificial supervision signal is imposed to predict the applied transformation on each image instance. Among this line of works, the representative ones consider mainly two classes of instance-wise transformations on input images. The first one is classifying global transformations, such as rotations [123], scaling and tiling [122], where the learning objective is to recognize the geometric transformation applied on an image. The second one is predicting local transformations, such as patch orderings



(a) predict rotation & input transformation



(b) predict patch ordering

Fig. 9: In instance-level pretext tasks (§3.2.1), **(a)** predict rotation or input transformation, **(b)** predict patch ordering, the aim is to predict the transformation on the input.

[11] and patch re-orderings [121], [162], [164], which cut each image into multiple local patches. The goal of patch orderings is to recognize the order of a given cut-out patch, while patch re-orderings, also known as the jigsaw puzzles, permute the cut-out patches randomly and the goal is to predict the permuted configurations. In summary, the objective of an instance-level pretext task can be abstractly written as:

$$\min_{\theta} \sum_{x \in \mathcal{D}} \mathcal{L}_{\text{unsup}}(\Phi_z(x), z, \theta), \tag{11}$$

where $\mathcal{L}_{\text{unsup}}(\cdot)$ can be various loss functions (e.g., cross-entropy loss [123]) that learn a mapping from a transformed input image $\Phi_z(x)$ to a discrete category or a configuration of the applied transformation z. Once trained, the representations are covariant with the transformations $\Phi_z(\cdot)$, thus being aware of the spatial context information, e.g., how an image is rotated or how the local patches are permuted.

Remarks. Although self-supervised learning objectives of pixel-level or instance-level pretext tasks are generally not explicitly related to the downstream task objectives (e.g., image classification, detection and segmentation), they permit to learn from unlabeled data by predicting the spatial context or structured correlation in images, such as inpainting missing regions, and predicting the applied rotations. As these self-supervision signals can implicitly uncover the semantic content (e.g. human interpretable concepts [166]) or spatial context in images, they often yield a meaningful pre-trained model for initialization in unseen downstream tasks, or even serve as a flexible and effective regularizer to facilitate other machine learning setups, such as semi-supervised learning [54] and domain generalization [167].

3.2.2 Discriminative Models

Discriminative models hereby refer to the class of unsupervised discriminative models that learn visual representations from the unlabeled data by enforcing invariance towards various task-irrelevant visual variations at either instance-level, neighbor-level or group-level. These visual variations can be intra-instance variations such as different views of the same instance [136], [168], [169], [170], [171], or inter-instance variations between neighbor instances [172], [173] or across a group of instances [140], [146], [174].

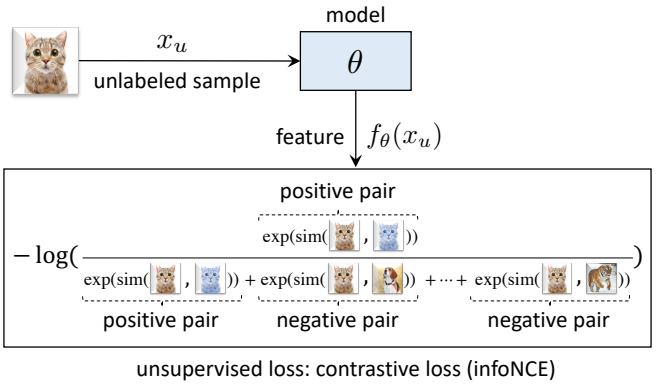


Fig. 10: Unsupervised discriminative model using contrastive learning (§3.2.2). The model is trained to pull together the positive pairs and push away the negative ones.

In the following, we review two representative classes of unsupervised discriminative models that offer the state of the art in unsupervised visual feature learning, including instance discrimination and deep clustering. The former imposes self-supervision per instance by treating each instance as a class, while the latter introduces supervision per group by considering a group of similar instances as a class. See Figure 10 and Figure 11 for the illustration of instance discrimination (based on contrastive learning) and deep clustering, which are also detailed in the following.

Instance discrimination refers to the class of models that learn discriminative representations by enforcing invariance towards different viewing conditions, data augmentations or various parts of the same image instance [12], [16], [55], [119], [120], [131], [135], [136], [168], [169], [170], [171], [175], [176], [177] – also known as exemplar learning [119], [120].

The most prevalent scheme in instance discrimination is **contrastive learning**, which was initially proposed to learn invariant representations by mapping similar inputs to nearby points in the latent space [105], [106]. The recent state-of-the-art contrastive learning models for self-supervised learning are generally proposed to obtain an invariance property by optimizing a contrastive loss formulated upon the noise contrastive estimation (NCE) principle [178], which maximizes the mutual information across different views. The multi-view information bottleneck model [179] extends the original information bottleneck principle to unsupervised learning and trains an encoder to retain all the relevant information for predicting the label while minimizing the excess information in the representation. Formally, contrastive learners such as SimLR [12] and MoCo [16] are generally optimized by an instance-wise contrastive loss (i.e., infoNCE loss) [106], [180]:

$$\min_{\theta} \sum_{x_i \in \mathcal{D}} -\log \frac{\exp(f_\theta(x_i) \cdot f_\theta(x_i^+)/\tau)}{\sum_{j=1}^{M} \exp(f_\theta(x_i) \cdot f_\theta(x_j)/\tau)}, \tag{12}$$

where τ is a temperature, f_θ is the feature encoder, i.e., a

DNN; $f_\theta(x_i)$, $f_\theta(x_i^+)$ are the feature embeddings of two different augmentations, or views of the same image; $\{x_j\}_{j=1}^M$ includes $(M-1)$ negative samples and 1 positive (i.e., x_i^+) sample. Eq. (12) optimizes the network by enforcing the positive pairs (i.e., embeddings of the same instance) to lie closer, while pushing apart the negative pairs (i.e., embeddings of different instances). Minimizing the InfoNCE loss is equivalent to maximizing a lower bound on the mutual information between $f_\theta(x_i)$ and $f_\theta(x_i^+)$ [168].

To derive a tractable yet meaningful contrastive distribution in Eq. (12), a large amount of negative pairs are often required per training batch. To this aim, existing state-of-the-art methods are typically featured with different *negative sampling* strategies to collect more negative pairs. For instance, a large batch size of 4096 is adopted in SimCLR [12]. In InstDis [131], MoCo [16], PIRL [135], and CMC [136], a memory bank is used to maintain all the instance prototypes by keeping moving average of their feature representations over training iterations. Finally, running queue enqueues the features of samples in the latest batches and dequeues the old mini-batches of samples to store a fraction of sample's features from the preceding mini-batches [16], [135], [177].

Inspired by deep metric learning, various training strategies are recently proposed to further boost contrastive learning. For instance, a hard negative sampling strategy [132] is introduced to mine the negative pairs that are similar to the samples but likely belong to different classes. Another line of works further propose to train negative pairs and (or) positive pairs by adversarial training [133], [134], which learns a set of "adversarial negatives" that are confused with the given samples, or trains the "cooperative positives" that are similar to the given samples. These strategies are designated to find the better negative and positive pairs for improving contrastive learning.

In addition to negative sampling, it is essential to apply various image transformations for generating multiple diverse variants (i.e., views) of the same instance to construct the positive pairs. The most typical way is to apply common data augmentation such as random cropping and color jittering [12], [16], [131], [135], [169], [176], or pretext transformation [135] like patch re-ordering [121] and rotation [123]. An alternative way is to artificially construct multiple views of a single image by using different image channels like luminance and chrominance [136], or by extracting the local and global patches of the same image [168]. In a nutshell, although there are different strategies in negative sampling and image transformations to construct the negative and positive pairs for contrastive learning, these strategies share the same aim to learn visual representations invariant to diverse input transformations [135], [175].

While **contrastive learning** approaches rely on obtaining a sufficient amount of negative pairs to derive the contrastive loss (Eq. (12)), another alternative **non-contrastive** scheme for instance discrimintation operates in a **negative-sample-free** manner [137], [138], [139], [181], as exemplified by bootstrap (BYOL) [138] and simple siamese networks (SimSiam) [137]). In particular, in BYOL and SimSiam, two views (obtained from data augmentation) of the same images are passed towards the networks and the mean squared error is minimized between the representations of two views to enforce invariances. Importantly, a stop gradient scheme

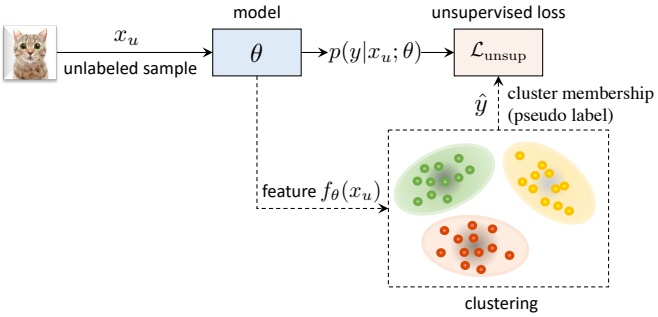


Fig. 11: In unsupervised discriminative models using deep clustering (§3.2.2), unlabeled samples are assigned to a set of clusters by *online* or *offline* clustering, while the cluster memberships are utilized as pseudo labels for training.

is adopted to prevent representational collapse, i.e. avoid mapping all the samples to the same representations. Another related method is Barlow Twins [181], which computes a cross-correlation matrix between the distorted versions of a batch of training samples and enforce the matrix to be an identity matrix, thus learning self-supervised representations invariant to different distortions. Although these non-contrastive methods adopt other loss formulations, they all share the similar spirit as contrastive learning given that meaningful representations are learned by enforcing invariances to different views of the same instance.

Deep clustering is another family of strong unsupervised models that learn meaningful representations by grouping similar instances from the same cluster together [140], [141], [142], [144], [146], [173], [174], [182], [183], [184], [185], [186], [187]. In training, the entire dataset is generally divided into groups by associating each instance to a certain cluster centroid based on pairwise similarities. Although clustering algorithms are longstanding machine learning techniques [188], [189], [190], they have been recently re-designed to be seamlessly integrated with DNNs to learn discriminative representations without label supervision. Conceptually, the cluster memberships can be considered as some pseudo labels to supervise the model training, as written in Eq. (13).

$$\min_\theta \sum_{x\in\mathcal{D}} \mathcal{L}_{\text{unsup}}(x, \hat{y}, \theta), \tag{13}$$

where $\hat{y}$ is the cluster membership of sample x, $\mathcal{L}_{\text{unsup}}(\cdot,\cdot,\theta)$ is the loss function that constrains the mapping from x to y, such as a classification loss. Deep clustering algorithms can be further grouped into two categories according to whether the assignments of cluster memberships are derived in an offline or online manner, as detailed in the following.

In offline clustering, unsupervised training is alternated between a cluster assignment step and a network training step [141], [142], [173], [174], [182], [191], [192], [193]. While the former step estimates the cluster memberships of all the training samples, the latter uses the assigned cluster memberships as pseudo labels to train the network. Representative offline clustering models include DeepCluster [140], JULE [141] and SeLa [142], which mainly differ in the clustering algorithms. Specifically, DeepCluster [140], [174] groups visual features using k-means clustering [189]. JULE [141] uses agglomerative clustering [194] that merges similar clusters to iteratively derive new cluster memberships.

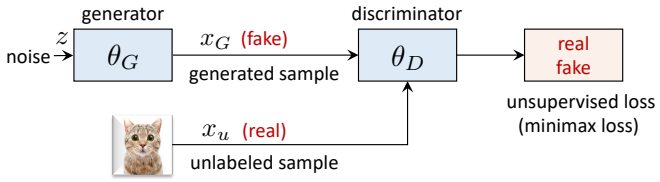


Fig. 12: In unsupervised deep generative model based on GAN (§3.2.3), a generator and a discriminator are trained with a minimax game (Eq. (14)) in an unsupervised manner, whilst their intermediate features lead to discriminative visual representations.

SeLa [142] casts clustering as an optimal transport problem solved by Sinkhorn-Knopp algorithm [195] to obtain the cluster memberships as pseudo labels.

In online clustering, the cluster assignment step and network training step are coupled in an end-to-end training framework, as represented by IIC [143], AssociativeCluster [145], PICA [144], and SwAV [146]. Compared to offline clustering, online clustering could better scale to large-scale datasets, as it does not require clustering the entire dataset iteratively. This is typically achieved in two ways: (1) training a classifier that parameterizes the cluster memberships (e.g., IIC and PICA); (2) learning a set of cluster centroids/prototypes (e.g., AssociativeCluster and SwAV). For instance, IIC [143] learns the cluster memberships by maximizing the mutual information between predictions of an original instance and a randomly perturbed instance obtained from data augmentation. SwAV [146] learns a set of prototypes (i.e., cluster centroids) in the feature space and assigns each sample to the closest prototype.

Remarks. Recent advances of discriminative unsupervised models include both contrastive learning and deep clustering, which have set the new state of the art. On one side, contrastive learning discriminates individual instances by imposing transformation invariance at the instance-level. Interestingly, this opposes some instance-level pretext tasks that instead learn by predicting the applied transformations. Contrastive learning also closely relates to consistency regularization in SSL in the sense of enforcing invariance to transformations, although different loss functions are often used. However, as shown in [137], a pairwise loss objective – often used for consistency regularization in SSL – can be also effective as contrastive loss (Eq. (12)). This suggests that the essential idea behind them is identical – imposing transformation invariance at instance level. Deep clustering, on the other hand, discriminates between groups of instances for discovering the underlying semantic boundaries, and enforces group-level invariance. The idea of consistency regularization is also adopted by several deep clustering methods [143], [144], conforming its more generic efficacy beyond SSL. Lastly, discriminative unsupervised learning can also be conducted at both instance-level and group-level to learn more powerful representations [186], [196].

3.2.3 Deep Generative Models

Deep generative models (DGMs), as introduced in §2.2.4, are inherent unsupervised learners that explicitly model the data distribution [109], [110], [197], [198]. DGMs are applicable for both semi-supervised and unsupervised learning, as most represented by Generative Adversarial Networks (GANs) [49], [148], [149], [151], [199]. A typical GAN contains a discriminator D to differentiate real and fake samples, and a generator G that can serve as an image encoder to capture the semantics in latent space, as trained by a minimax game:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))], \tag{14}$$

where z is sampled from an input noise distribution $p_z(z)$. GANs can learn representations at both the discriminator and the generator level. See Figure 12 for an illustration of deep generative model based on a GAN.

To learn representations at the **discriminator-level**, Deep Convolutional Generative Adversarial Network (DCGAN) [147] adopts a pre-trained convolutional discriminator to extract features for tackling a downstream image classification task. Later on, Self-supervised GAN [148] and Transformation GAN [149] further imbue the discriminator with a self-supervised pretext task to predict the applied image transformation, thus enabling the representations to capture the latent visual structures.

To learn representations at the **generator-level**, Bidirectional Generative Adversarial Networks (BiGAN) [199] introduces an image encoder coupled with the generator, which is trained with a joint discriminator loss to tie the data distribution and the latent feature distribution together. This allows the image encoder to capture the semantic variations in its latent representation, and offer discriminative visual representations for one nearest neighbor (1NN) classification. To further improve BiGAN, BigBiGAN [151] adopts more powerful discriminator and generator architectures than BigGAN [150], together with an additional unary discriminator loss to constrain the data or latent distribution independently, therefore enabling more expressive unsupervised representation learning at the generator-level.

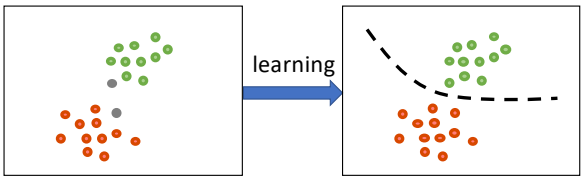
Remarks. Although most state-of-the-art UL methods are self-supervised models that solve pretext tasks or perform unsupervised discriminative learning (as reviewed in §3.2.1 and §3.2.2), deep generative models are still an important class of unsupervised learners owing to their native unsupervised nature to learn expressive data representations in a probabilistic manner. Further, they do not require manual design of a meaningful discriminative learning objective, while offering a unique ability to generate abundant data.

4 Discussion on SSL and UL

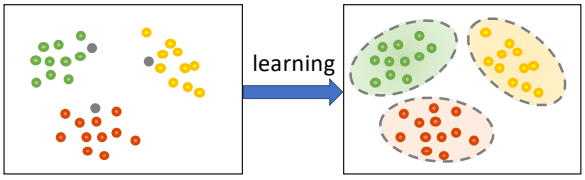
In this section, we reveal the connections between SSL and UL with further analysis and discussion, including their common learning assumptions (§4.1), and their applications in different computer vision tasks (§4.2).

4.1 The learning assumptions shared by SSL and UL

As discussed in §2.1, the unsupervised learning objectives in SSL are often formulated based on the smoothness assumption [70]. Broadly speaking, the learning assumptions of various discriminative SSL and UL algorithms can be grouped into two types of smoothness assumptions, i.e.



(a) local smoothness



(b) global smoothness

Fig. 13: The learning assumptions shared by SSL and UL: **(a)** local smoothness, and **(b)** global smoothness. During training, the grey dots (unlabeled samples) are assigned to certain class labels based on the decision boundaries derived from the local or global smoothness assumption.

local smoothness and global smoothness – as visually illustrated in Figure 13. In the following, we further elaborate these assumptions and discuss the different SSL and UL algorithms that are built upon these assumptions.

4.1.1 Local Smoothness

The **local smoothness** is often assumed in two ways to learn from the unlabeled samples. First, a sample x_i is assumed to share the same class label as its transformed variant $\hat{x}_i$ (Eq. (15)). Second, a sample x_i is assumed to belong to the same class as its nearby sample x_j in the latent representation space (Eq. (16)). The following equations define how an unsupervised loss term is imposed to enforce local smoothness on an unlabeled sample x_i:

$$\min_{\theta} \sum_{x_i \in \mathcal{D}} \mathcal{L}_{\text{unsup}}(f(x_i), f(\hat{x}_i)) \tag{15}$$

$$\min_{\theta} \sum_{x_i \in \mathcal{D}} \mathcal{L}_{\text{unsup}}(f(x_i), f(x_j)) \tag{16}$$

where $f(\cdot)$ is the model to be trained and gives the model output (such as features or predictions). $\mathcal{L}_{\text{unsup}}(\cdot)$ could be any similarity metric that quantifies the divergence or inconsistency between two model outputs, such as a mean square error, or contrastive loss. In essence, multiple SSL and UL algorithms are formulated based on the local smoothness to learn from the unlabeled samples, as detailed below.

To impose the local smoothness among different transformed views of the same samples (Eq. (15)), the consistency regularization techniques in SSL (§2.2.1, Figure 3) enforce the predictive smoothness of the same samples under different variations imposed at the input space and (or) model space, given that the different transformed versions of the same sample should lie in its own local neighborhood. Similarly, the instance discrimination algorithms in UL also implicitly enforce the same samples under different views or transformations to have locally consistent representations, as represented by contrastive learning which encourages local invariances on each sample (§3.2.2, Figure 10).

To impose the local smoothness among the nearby samples (Eq. (16)), the graph-based regularization techniques in SSL often propagate the class labels to the unlabeled samples using the labels of their neighbours on the graph, as the nearby samples should likely share the same class (§2.2.3, Figure 5). Similarly, neighbourhood consistency is also explored in UL [172], [173], which forms the semantic training labels by mining the nearest neighbors of each sample based on feature similarity, given that nearest neighbors are likely to belong to the same semantic class.

TABLE 3: A summarized taxonomy on the related SSL and UL algorithms based on their learning assumptions.

Assumption	Objective	Corresponding SSL & UL algorithms
local smoothness	Eq. (15)	consistency regularization in SSL (§2.2.1) instance discrimination in UL (§3.2.2)
	Eq. (16)	graph-based regularization in SSL (§2.2.3) neighbourhood consistency in UL (§3.2.2)
global smoothness	Eq. (17)	self-training in SSL (§2.2.2) deep clustering in UL (§3.2.2)

4.1.2 Global Smoothness

The **global smoothness** considers that a sample x_i could be assigned to a certain class (or target) z_i based on the underlying global structures captured by the model (Eq. (17)).

$$\min_{\theta} \sum_{x_i \in \mathcal{D}} \mathcal{L}_{\text{unsup}}(f(x_i), z_i) \tag{17}$$

where z_i is the learning target (e.g. the cluster membership or the most confident predicted class), which is derived from the global class decision boundaries discovered during training (Figure 13) whilst the decision boundaries are supposed to lie in low density regions. Similar to Eq. (15) and Eq. (16), $\mathcal{L}_{\text{unsup}}(\cdot)$ is a similarity metric that quantifies the inconsistency between the model output and the training target, such as a cross-entropy loss. The global smoothness assumption is also widely adopted in various SSL and UL techniques to learn from the unlabeled samples with pseudo learning targets, as detailed in the following.

The self-training techniques in SSL (§2.2.2, Figure 4) are generally formulated based on global smoothness, as the learning targets for unlabeled data are derived based on the class decision boundaries discovered by the models. For instance, in entropy minimization (Eq. (4), Figure 4 (a)), the pseudo label is obtained as the class predicted with the highest confidence. In co-training and distillation (Eq. (5), Eq. (6), Figure 4 (b)(c)), the learning targets come from the model co-trained in parallel or pre-trained beforehand. Similarly, the deep clustering algorithms in UL (§3.2.2, Figure 11) are also proposed upon global smoothness, given that the cluster memberships for unlabeled samples are acquired from an online or offline clustering algorithm which uncovers the latent class decision boundaries in the feature space.

4.1.3 Connections between SSL and UL

The shared learning rationales. As analyzed in §4.1.1 and §4.1.2, most SSL and UL algorithms are formulated based on the same local smoothness or global smoothness assumption – as summarized in Table 3. A common aspect of these SSL and UL algorithms is to design visual learning objectives that enforce invariance or equivariance towards different transformations applied on the input data, as

represented by consistency regularization in SSL (§2.2.1) and instance discrimination (such as contrastive learning) in UL (§3.2.2). Typical transformation strategies can range from simple data augmentation [21], [22], [35], to more complex transformations such as adversarial perturbations [24], [25], [74], [81], rotations [123] and patch reordering [121], autoencoding transformations [200], [201] and automated augmentation [28], [29], [30]. On one side, most of these SSL and UL methods hinge on learning representations that are invariant to different transformations including data augmentation and perturbations by assigning the same underlying labels to the augmented and perturbed data samples. On the other side, other SSL and UL methods consider learning representations that are equivalent to different transformations such as rotations and patch reordering by learning to predict the types of transformations.

In summary, many state-of-the-art SSL and UL methods can be well related with the same underlying learning assumptions, given that they introduce quite similar objectives to learn from the unlabeled samples. In essence, the learning rationales of these SSL and UL methods could be broadly categorized as three types: (1) impose the consistency among different transformed versions of the same samples (Eq. (15)), (2) enforce the smoothness between a sample and its neighbouring sample (Eq. (16)), and (3) derive the learning targets for the unlabeled samples based on the global decision boundaries (Eq. (17)).

The similarities and differences in problem setups. In the problems setups, SSL and UL are similar in the sense that both labeled and unlabeled data are often involved in their training protocols before evaluating their generalized model performance on the test set. In particular, the SSL paradigm adopts ***one-stage*** training and uses both labeled and unlabeled data during training (Figure 2); while most existing UL protocols consider ***two-stage*** training (Figure 7) – one stage for ***pre-training*** with unlabeled data and another stage for ***fine-tuning*** with labeled data on a *downstream task*.

We summarize the training protocols used for SSL and UL in Table A in the supplementary. In brief, UL differs from SSL in several ways: (1) the labeled data and unlabeled data are not given together at once; (2) unlabeled set may have a different distribution from the labeled one. All these properties make UL a more generic learning paradigm to leverage different unlabeled datasets. Nevertheless, how unsupervised pre-training upon different forms of unlabeled data benefits the model generalization on specific downstream tasks remains an open research question. For instance, it remains unclear how an unsupervised model pre-trained on natural colour images could generalize to a downstream task that has a different data distribution such as grayscale images in medical imaging. In this regard, SSL provides a more reliable learning paradigm to utilize the unlabeled data, given that the label set offers the prior knowledge for the models and (or) the model designers to select the useful set of unlabeled samples that are similar to the labeled data distribution.

4.2 Applied SSL and UL in Visual Recognition

In §2 and §3, we mainly present the SSL and UL algorithms for standard image classification. However, their underlying learning rationales could be generalized to tackle other challenging computer vision tasks, e.g., semantic segmentation [65], [202], object detection [63], [203], unsupervised domain adaptation [204], [205], pose estimation [67], [206], 3D scene understanding [207], video recognition [152], [208], etc. In the following, we review three core visual recognition tasks that widely benefit from SSL and UL methods to exploit unlabeled data, including semantic segmentation (§4.2.1), object detection (§4.2.2), and unsupervised domain adaptation (§4.2.3). We also give a taxonomic overview of different methods for different tasks in Table B in the supplementary.

4.2.1 Semantic Segmentation

Semantic segmentation aims to assign a semantic class label for each pixel an input image. It is a core computer vision task that could benefit various real-world applications such as medical image analysis [209], [210], [211], [212] and autonomous driving [213], [214], [215]. Supervised semantic segmentation requires tedious and expensive pixel-wise label annotations, e.g. manually annotating one single natural image in Cityscapes needs 1.5 hour [213].

To reduce the annotation costs in semantic segmentation, a group of recent works consider only a small set of the training data is annotated with per-pixel semantic labels while the rest of the training data is unlabeled – known as **semi-supervised semantic segmentation**. These works generally inherit similar learning rationales as SSL or UL for image classification, and adapt techniques such as consistency regularization [216], [217], [218], [219], self-training [202], [210], [220], [221], [222], [223], [224], GAN frameworks [225], [226], [227] in SSL, or contrastive learning [228], [229], [230], [231] in UL to learn from unlabeled images. Nevertheless, unsupervised loss terms in semantic segmentation are often required to impose in a per-pixel manner to align with the pixel-wise learning objective in semantic segmentation. In the following, we discuss the three most representative lines of state-of-the-art methods driven by recent advances in SSL and UL for semi-supervised semantic segmentation.

Consistency regularization (§2.2.1) can be generalized for pixel-wise tasks by formulating the consistency loss (Eq. (2), Eq. (3)) at the pixel level. Rooted in similar spirit as standard consistency regularization in SSL, recent works in semi-supervised semantic segmentation [216], [217], [218], [219] resort to enforcing pixel consistency among the images before and after perturbations, whilst perturbations can be introduced at the input space [216] or feature space [217]. For instance, the first consistency regularization method for semantic segmentation [216] applies CutOut [78] and CutMix [232] augmentation techniques to perturb the input images with partial corruption, and imposes pixel-level loss terms to ensure the uncorrupted regions in perturbed images should have consistent pixel-wise predictions as the same regions in original images. A recent cross-consistency training [217] instead applies feature perturbations by injecting noise into network's activations and enforces pixel consistency between the clean and perturbed outputs.

Self-training algorithms (§2.2.2) are adapted and shown effective for semi-supervised semantic segmentation [202], [210], [220], [221], [222], [223], [224], where pseudo segmentation maps on unlabeled images are propagated using a pre-trained teacher model [223], or a co-trained model [202].

For example, a recent self-training method [223] propagates pseudo segmentation labels with two steps – (1) assigning pixel-wise pseudo labels on unlabeled data with a pre-trained teacher model; and (2) re-training a student model with the re-labeled dataset – until no more performance gain is achieved. Another self-training approach [202] adopts a co-training scheme by training two models to learn the per-pixel segmentation predictions from each others.

Contrastive learning is widely used in UL (§3.2.2) and recently adapted to learn from unlabeled data in semantic segmentation [228], [229], [230], [231]. To formulate the contrastive loss (Eq. (12)) per pixel, one needs select meaningful positive and negative pairs with consideration of pixel spatial locations. For this aim, a directional context-aware contrastive loss [228] is proposed to crop two patches from one image, and take features at the same location as a positive pair and the rest as negative pairs. Another pixel contrastive loss [230] is introduced to align the features before and after a random color augmentation by taking features at the same location as a positive pair, while sampling a fix amount of negative pairs from different images.

4.2.2 Object Detection

Object detection aims to predict a set of bounding boxes and the corresponding class labels for the objects of interest in an image. An object detector needs to unify classification and localization into one model by jointly training a classifier to predict class labels and a regression head to generate the bounding boxes [5], [233]. It is an important computer vision task that widely impacts different applications such as person search [234], vehicle detection [235], logo detection [236], text detection [237], etc. Supervised object detection requires costly annotation efforts – annotating the bounding box of a single object could take up to 42 seconds [238].

To exploit the unlabeled data without bounding box or class label information, a group of recent works in object detection exploit unlabeled data to boost model generalization by training on a small set of labeled data and a set of completely unlabeled images – known as **semi-supervised object detection**. These works mainly reformulate two streams of SSL techniques, including consistency regularization [63], [203], [239], [240], [241], [242] and self-training [42], [243], [244], [245], [246], both of which introduce the learning targets for both bounding boxes and class labels to learn from the completely unlabeled data, as detailed next.

Consistency regularization (§2.2.1) is introduced for semi-supervised object detection to propagate the soft label and bounding boxes assignment on unlabeled images based on dual consistency constraints on classification and regression [63], [203], [239], [240], [241], [242]. One line of works apply data augmentation such as random flipping [203] and MixUp [75] to generate augmented views of unlabeled images and encourage the predicted bounding boxes and its class labels remain consistent for the different views. Compared to standard consistency regularization, these methods especially need re-estimating the bounding box location in an augmented image, such as flip the bounding box [203], or calculate the overlapped bounding boxes of two mixed images in MixUp [75]. Another line of works follow a teacher-student training framework and impose teacher-student consistency [63], [240], [241], [242] similar to Mean Teacher [35]. The teacher model is derived either from the student model via exponential mean average (EMA) [63], [240], [242], or by applying non-maximum suppression (NMS, a filtering technique for refining the detected bounding boxes) on the instant model outputs [241] to obtain the pseudo bounding boxes and label annotations for training.

Self-training algorithms (§2.2.2) are also introduced to annotated unlabeled images for object detection [42], [243], [244], [245], [246]. A simple self-training paradigm is to annotate the unlabeled images with bounding boxes and their class labels using a pre-trained teacher model and use these data for re-training [243]. However, such pseudo annotations may be rather noisy. To improve the quality of pseudo labels, recent works propose interactive self-training to progressively refine the pseudo labels with NMS [244], or quantify model uncertainty to select or derive more reliable pseudo labels [245], [246] to learn from unlabeled data.

4.2.3 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) can be deemed as a special case of SSL where the labeled (source) and unlabeled (target) data lie in different distributions, a.k.a. different domains. UDA is essential for visual recognition [247], as the statistical properties of visual data are sensitive to a wider variety of factors, e.g., illumination, viewpoint, resolution, occlusion, times of the day, and weather conditions. While most UDA methods focus on tackling the domain gap between the labeled and unlabeled data, SSL and UL algorithms can also be adapted to learn from unlabeled data in UDA, as detailed in the following.

Consistency regularization (§2.2.1) is shown effective in UDA. Rooted in the same spirit of encouraging consistent outputs under perturbations, various UDA approaches apply input transformations or model ensembling to simulate variations in input or model space [35], [248], [249], [250]. To generate input variations, a dual MixUp regularization integrates category-level MixUp and domain-level MixUp to regularize the model with consistency constraints, thus learning from unlabeled data to enhance domain-invariance [248]. To generate model variations, self-ensembling [249] utilizes the Mean Teacher [35] to impute training targets on the unlabeled training data in target domain.

Self-training (§2.2.2) has been also useful for UDA. Similar to SSL, self-training for UDA include three streams of techniques to impute pseudo labels on the unlabeled target samples, including entropy minimization, pseudo-label and co-training. To ensure the effectiveness, self-training methods are often coupled with domain distribution alignment for reducing the domain shift. For instance, entropy minimization (Eq. (4)) is adopted for UDA [251], [252], [253], in combination with distribution alignment techniques such as domain-specific batch normalization layers [251], aligning second-order statistics of features [252], or adversarial training and gradient synchronization [253]. Co-training (Eq. (5)) is also introduced for UDA, which imputes training targets from multiple co-trained classifiers to learn from unlabeled data and match cross-domain distributions [254].

Deep generative models (DGMs), as a class of models for SSL and UL (§2.2.4, §3.2.3), are widely adopted for UDA. In contrast to other UDA methods that reduce the domain shift at the feature level, DGMs provide an alternative and

complementary solution to mitigate the domain discrepancy at pixel level by cross-domain image-to-image translation. The majority of these frameworks are based on GANs, such as PixelDA [255], generate to adapt [256], and GANs with cycle-consistency like CyCADA [257], SBADA-GAN [258], I2I Adapt [259] and CrDoCo [260]. These models typically learn a real-to-real [257], [258], [260], [261] or synthetic-to-real [255], [256], [262] mapping to render the image style from the labeled source to the unlabeled target domain, thus offering synthetic training data with pseudo labels.

Self-supervised learning, which has been popularized in SSL and UL (§2.2.5, §3.2.1), is also introduced in UDA to construct auxiliary self-supervised learning objectives on unlabeled data. Self-supervised models often address the UDA problem by self-supervision coupled with a supervised objective on the labeled source data [167], [204], [263], [264], [265]. The pioneer work in this direction is JiGen [167], which learns jointly to classify objects and solve the jigsaw puzzles [121] pretext task to achieve better generalization in new domains. Recent works [204], [263], [264] explored other self-supervised pretext tasks such as predicting rotation [204], [263], [264], flipping [204] and patch ordering [204]. Besides pretext tasks, recent UDA methods also explored discriminative self-supervision signals based on clustering or contrastive learning. For instance, Domain Adaptative Neighborhood Clustering via Entropy optimization (DANCE) [205] performs neighborhood clustering by assigning the target samples to a "known" class prototype in the source domain or its neighbor in the target domain. Gradient regularized contrastive learning [266] leverages the contrastive loss to push the unlabeled target samples towards the most similar labeled source samples. Similarly, a recent cross-domain contrastive learning [265] approach aligns the target domain features to the class prototype features in the source domain through contrastive loss, thus minimizing the distances between the cross-domain samples that likely belong to the same class.

5 Emerging Trends and Open Challenges

In this section, we discuss the emerging trends in semi-supervised and unsupervised learning from unlabeled data, covering three directions, namely open-set learning (§5.1), incremental learning (§5.2) and multi-modal learning (§5.3). We detail both recent developments and open challenges.

5.1 Open-Set Learning from Unlabeled Data

In §2, we review works addressing the relatively simple closed-set learning in SSL, which assume that unlabeled data share the same label space as the labeled one. However, this closed-set assumption may greatly hinder the effectiveness of SSL in leveraging real-world uncurated unlabeled data that contains unseen classes, i.e., out-of-distribution (OOD) samples (also known as outliers) [69]. When applying most existing SSL methods to open-set learning with noisy unlabeled data, their model performance may degrade significantly, as the OOD samples could induce catastrophic error propagation.

A recent line of works propose to address a more complex *open-set SSL* scenario [14], [15], [267], [268], [269], [270], [271], [272], where the unlabeled set contains task-irrelevant OOD data. In this setup (so-called open-world SSL), unlabeled samples are not all beneficial. To prevent possible performance hazards caused by unlabeled OOD samples, recent advances in SSL propose various sample-specific selection strategies to discount their importance or usage [14], [15], [267], [268]. The pioneer works including UASD [14] and DS^{3}L [15] propose to impose a dynamic weighting function to down-weight the unsupervised regularization loss term proportional to the likelihood that an unlabeled sample belongs to an unseen class. Follow-up works resort to curriculum learning [267] and iterative self-training [268] by training an OOD classifier to detect and discard the potentially detrimental samples. More recently, OpenMatch [270] propose to train a set of one-vs-all classifiers for detecting inliers and outliers and regularize the model with a consistency constraint on the unlabeled inliers.

Open Challenges. The open-set setup in SSL calls for integrating OOD detection [273] or novel class discovery [274] with semi-supervised learning in a unified model to advance selective exploitation of noisy unlabeled data. Moreover, a more recent work propose a universal SSL benchmark [271] which further extends the distribution mismatch problem in open-set setup as subset or intersectional class mismatch, and feature distribution mismatch. These more realistic setups pose multiple new challenges, including confidence calibration of DNN for OOD detection [273], [273], [275], [276], [277], imbalanced class distribution caused by real-world long-tailed distributed unlabeled data [278], [279], and discovery of unseen classes in unlabeled data [274], [280], [281]. Although recent advances in open-set SSL have explored OOD detection, the other challenges remain to be resolved to exploit real-world unlabeled data.

5.2 Incremental Learning from Unlabeled Data

Existing works on SSL and UL often assume all unlabeled training data is available at once, which however may not always hold in practice due to privacy concerns or computational constraints. In many realistic scenarios, we need to perform *incremental learning* (IL) with new data to update the model incrementally without access to past training data. Here we review recent research directions on IL from unlabeled data [282], [283] and discuss its open challenges.

Incremental learning (IL) from unlabeled data has been investigated in a semi-supervised fashion [282]. IL (also known as continual learning and lifelong learning [284]) aims to extend an existing model's knowledge without accessing the previous training data. Most existing IL approaches use regularization objectives to not forget old knowledge, i.e., reducing catastrophic forgetting [285], [286], [287], [288]. To this aim, unlabeled data is often used in IL to prevent catastrophic forgetting by estimating the importance weights of model parameters for old tasks [289], or formulating a knowledge distillation objective [282], [290] to consolidate the knowledge learned from old data. Recently, multiple works explore IL from unlabeled data that comes as a non-stationary stream [283], [291], with the class label space possibly varying over time [292]. In this setting, the goal is to learn a salient representation from continuous incoming unlabeled data stream. To expand the representations for novel classes and unlabeled data, several strategies

are adopted to dynamically update representations in the latent space, such as creating new cluster centroids by online clustering [292] and updating mixture-of-Gaussians [283]. Some recent works apply self-supervised techniques on the unlabeled test-data [293], [294], [295], which is useful to overcome possible shifts in the data distribution [296].

Open Challenges. Incremental learning from unlabeled data requires solving multiple challenges, ranging from catastrophic forgetting [282], [297], modeling new concepts [283], [292] to predicting the evolution of data streams [296]. Due to lacking the access to all the unlabeled training data at once, addressing these challenges is nontrivial as directly applying many existing SSL and UL methods could not guarantee good generalization performance. As an example, pseudo labels may suffer the confirmation bias problem [298] when classifying unseen unlabeled data. Thus, incremental learning from a stream of potentially non-*i.i.d.* unlabeled data remains an open challenge.

5.3 Multi-Modal Learning from Unlabeled Data

A growing amount of recent works bring non-visual modalities (e.g., text, audio) and visual modality to form meaningful self-supervision signals that enable learning from multi-modal unlabeled data. To bring vision and language for unsupervised learning, variants of vision and language BERT models (e.g., ViLBERT [299], LXMERT [300], VL-BERT [301], Uniter [302] and Unicoder-VL [303]) are built upon the transformer blocks [304] to jointly model images and natural language in an unsupervised way. Specifically, the visual, linguistic or their joint representations can be learned in an unsupervised manner by solving the *Cloze task* in natural language processing which predicts the masked words in the input sentences [305], or by optimizing a linguistic-visual alignment objective [300], [306]. Another line of works utilize the language supervision (e.g., from web data [307] or narrated materials [308], [309], [310], [311], [312], [313]) to guide unsupervised representation learning by aligning images and languages in the shared latent space, as exemplified by CLIP [312] and ALIGN [313].

Similarly, to bring audio and visual modalities for unsupervised learning, existing works exploit the natural audio-visual correspondence in videos to formulate various self-supervised signals, which predict the cross-modal correspondence [314], [315], align the temporally corresponding representations [309], [316], [317], [318], or cluster their representations in a shared audio-visual latent space [208], [319]. Several recent advances further explore audio, vision and language together for unsupervised representation learning by aligning different modalities in a shared multi-modal latent space [310], [320] or in a hierarchical latent space for audio-vision and vision-language [308].

Open Challenges. The success of multi-modal learning from unlabeled data often relies on an assumption that different modalities are semantically correlated. For instance, when clustering audio and video data for unsupervised representation learning [208], or transferring text knowledge to the unlabeled image data [321], the two data modalities are assumed to share similar semantics. However, this assumption may not hold in real-world data, leading to degraded model performance [309], [322]. Thus, it remains an unsolved challenge to learn from the multi-modal unlabeled data that contains a semantic gap across modalities.

6 Conclusion

Learning visual representations with limited or no manual supervision is critical for scalable computer vision applications. Semi-supervised learning (SSL) and unsupervised learning (UL) models provide feasible and promising solutions to learn from unlabeled visual data. In this comprehensive survey, we have introduced unified problem definitions and taxonomies to summarize and correlate a wide variety of recent advanced and popularized SSL and UL deep learning methodologies for building superior visual classification models. We believe that our concise taxonomies of existing algorithms and extensive discussions of emerging trends help to better understand the status quo of research in visual representation learning with unlabeled data, as well as to inspire new learning solutions for major unresolved challenges involved in the limited-label regime.

Acknowledgments

This work has been partially funded by the ERC (853489-DEXIM) and the DFG (2064/1–Project number 390727645).

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 2015.
[2] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *MIT press*, 2016.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.
[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.
[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.
[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE TPAMI*, 2017.
[7] Z.-H. Zhou and M. Li, "Semi-supervised regression with co-training." in *IJCAI*, 2005.
[8] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning," *IEEE TNNLS*, 2009.
[9] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *IJCV*, 2006.
[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE TPAMI*, 2013.
[11] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015.
[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.
[13] X. J. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2005.
[14] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch." in *AAAI*, 2020.
[15] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *ICML*, 2020.
[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.
[17] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *ML*, 2020.

[18] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE TPAMI*, 2020.

[19] L. Schmarje, M. Santarossa, S.-M. Schröder, and R. Koch, "A survey on semi-, self-and unsupervised learning for image classification," *arXiv preprint arXiv:2002.08721*, 2020.

[20] G.-J. Qi and J. Luo, "Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods," *IEEE TPAMI*, 2020.

[21] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *NeurIPS*, 2016.

[22] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *ICLR*, 2017.

[23] X. Wang, D. Kihara, J. Luo, and G.-J. Qi, "Enaet: A self-trained framework for semi-supervised and supervised learning with ensemble transformations," *IEEE TIP*, 2020.

[24] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," in *ICLR*, 2016.

[25] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE TPAMI*, 2018.

[26] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *NeurIPS*, 2019.

[27] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," in *IJCAI*, 2019.

[28] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *ICLR*, 2020.

[29] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *NeurIPS*, 2020.

[30] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020.

[31] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *NeurIPS*, 2014.

[32] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *NeurIPS*, 2015.

[33] S. Park, J.-K. Park, S.-J. Shin, and I.-C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *AAAI*, 2018.

[34] L. Zhang and G.-J. Qi, "Wcp: Worst-case perturbations for semi-supervised deep learning," in *CVPR*, 2020.

[35] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[36] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, "There are many consistent explanations of unlabeled data: Why you should average," in *ICLR*, 2019.

[37] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *ICMLW*, 2013.

[38] Y. Chen, X. Zhu, and S. Gong, "Semi-supervised deep learning with memory," in *ECCV*, 2018.

[39] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *ECCV*, 2018.

[40] W. Dong-DongChen and Z.-H. WeiGao, "Tri-net for semi-supervised deep learning," in *IJCAI*, 2018.

[41] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020.

[42] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," in *CVPR*, 2018.

[43] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *ICML*, 2008.

[44] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," in *CVPR*, 2018.

[45] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[46] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *CVPR*, 2019.

[47] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *NeurIPS*, 2014.

[48] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," in *ICML*, 2016.

[49] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," in *ICLR*, 2016.

[50] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NeurIPS*, 2016.

[51] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially learned inference," in *ICLR*, 2017.

[52] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *NeurIPS*, 2017.

[53] G.-J. Qi, L. Zhang, H. Hu, M. Edraki, J. Wang, and X.-S. Hua, "Global versus localized generative adversarial nets," in *CVPR*, 2018.

[54] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *ICCV*, 2019.

[55] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *NeurIPS*, 2020.

[56] R. Fergus, Y. Weiss, and A. Torralba, "Semi-supervised learning in gigantic image collections," in *NeurIPS*, 2009.

[57] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *ICLR*, 2017.

[58] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998.

[59] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *CIKM*, 2000.

[60] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nature Reviews Genetics*, 2015.

[61] Y. K. Jang and N. I. Cho, "Generalized product quantization network for semi-supervised image retrieval," in *CVPR*, 2020.

[62] J. Gao, J. Wang, S. Dai, L.-J. Li, and R. Nevatia, "Note-rcnn: Noise tolerant ensemble rcnn for semi-supervised object detection," in *ICCV*, 2019.

[63] Y. Tang, W. Chen, Y. Luo, and Y. Zhang, "Humble teachers teach better students for semi-supervised object detection," in *CVPR*, 2021.

[64] T. Kalluri, G. Varma, M. Chandraker, and C. Jawahar, "Universal semi-supervised semantic segmentation," in *ICCV*, 2019.

[65] Y. Ouali, C. Hudelot, and M. Tami, "Semi-supervised semantic segmentation with cross-consistency training," in *CVPR*, 2020.

[66] M. S. Ibrahim, A. Vahdat, M. Ranjbar, and W. G. Macready, "Semi-supervised semantic image segmentation with self-correcting networks," in *CVPR*, 2020.

[67] Y. Chen, Z. Tu, L. Ge, D. Zhang, R. Chen, and J. Yuan, "So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning," in *ICCV*, 2019.

[68] R. Mitra, N. B. Gundavarapu, A. Sharma, and A. Jain, "Multiview-consistent semi-supervised learning for 3d human pose estimation," in *CVPR*, 2020.

[69] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *NeurIPS*, 2018.

[70] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NeurIPS*, 2004.

[71] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *NeurIPS*, 2002.

[72] O. Chapelle, A. Zien, C. Z. Ghahramani *et al.*, "Semi-supervised classification by low density separation," in *AISTATSW*, 2005.

[73] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," *Technical Report, Carnegie Mellon University*, 2002.

[74] T. Suzuki and I. Sato, "Adversarial transformations for semi-supervised learning." in *AAAI*, 2020.

[75] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[76] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[77] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *CVPRW*, 2020.

[78] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[79] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[80] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR*, 2017.

[81] A. Najafi, S.-i. Maeda, M. Koyama, and T. Miyato, "Robustness to adversarial perturbations in learning from incomplete data," in *NeurIPS*, 2019.

[82] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," in *NeurIPS*, 2019.

[83] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, "Fast autoaugment," in *NeurIPS*, 2019.

[84] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population based augmentation: Efficient learning of augmentation policy schedules," in *ICML*, 2019.

[85] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, "Adversarial autoaugment," in *ICLR*, 2019.

[86] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *UAI*, 2018.

[87] T. M. Mitchell, "Generalization as search," *AI*, 1982.

[88] R. E. Schapire, "The strength of weak learnability," *ML*, 1990.

[89] L. Breiman, "Random forests," *ML*, 2001.

[90] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, 1999.

[91] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NeurIPS*, 2017.

[92] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *NeurIPS*, 2005.

[93] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Mutual exclusivity loss for semi-supervised deep learning," in *ICIP*, 2016.

[94] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, 2017.

[95] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. W. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning," in *CVPR*, 2019.

[96] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[97] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[98] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *ACM SIGKDD*, 2006.

[99] J. Ba and R. Caruana, "Do deep nets really need to be deep?" *NeurIPS*, 2014.

[100] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan, "Billion-scale semi-supervised learning for image classification," *arXiv preprint arXiv:1905.00546*, 2019.

[101] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.

[102] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *JMLR*, 2006.

[103] B. Wang, Z. Tu, and J. K. Tsotsos, "Dynamic label propagation for semi-supervised multi-class multi-label classification," in *CVPR*, 2013.

[104] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-supervised learning with graph learning-convolutional networks," in *CVPR*, 2019.

[105] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *IJPRAI*, 1993.

[106] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006.

[107] W. Lin, Z. Gao, and B. Li, "Shoestring: Graph-based semi-supervised classification with severely limited labeled data," in *CVPR*, 2020.

[108] S. Li, B. Liu, D. Chen, Q. Chu, L. Yuan, and N. Yu, "Density-aware graph for deep semi-supervised visual recognition," in *CVPR*, 2020.

[109] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[110] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[111] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[112] M. Ehsan Abbasnejad, A. Dick, and A. van den Hengel, "Infinite variational autoencoder for semi-supervised learning," in *CVPR*, 2017.

[113] A. Kumar, P. Sattigeri, and T. Fletcher, "Semi-supervised learning with gans: Manifold invariance with improved inference," in *NeurIPS*, 2017.

[114] C. Li, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *NeurIPS*, 2017.

[115] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis." in *ICML*, 2016.

[116] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw." in *NIPS*, 2016.

[117] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *CVPR*, 2018.

[118] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "F-vaegan-d2: A feature generating framework for any-shot learning," in *CVPR*, 2019.

[119] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *NeurIPS*, 2014.

[120] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE TPAMI*, 2015.

[121] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *ECCV*, 2016.

[122] M. Noroozi, H. Pirsiavash, and P. Favaro, "Representation learning by learning to count," in *ICCV*, 2017.

[123] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.

[124] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, 2006.

[125] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *ICANN*, 2011.

[126] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.

[127] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008.

[128] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *ECCV*, 2016.

[129] ——, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *CVPR*, 2017.

[130] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *CVPR*, 2017.

[131] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *CVPR*, 2018.

[132] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in *ICLR*, 2021.

[133] Q. Hu, X. Wang, W. Hu, and G.-J. Qi, "Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries," in *CVPR*, 2021.

[134] X. Wang, Y. Huang, D. Zeng, and G.-J. Qi, "Caco: Both positive and negative samples are directly learnable via cooperative-adversarial contrastive learning," *arXiv preprint arXiv:2203.14370*, 2022.

[135] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *CVPR*, 2020.

[136] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *ECCV*, 2020.

[137] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021.

[138] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar

et al., "Bootstrap your own latent: A new approach to self-supervised learning," in *NeurIPS*, 2020.
[139] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *ICML*, 2021.
[140] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018.
[141] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *CVPR*, 2016.
[142] Y. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *ICLR*, 2020.
[143] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *CVPR*, 2019.
[144] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *CVPR*, 2020.
[145] P. Haeusser, J. Plapp, V. Golkov, E. Aljalbout, and D. Cremers, "Associative deep clustering: Training a classification network with no labels," in *GCPR*, 2018.
[146] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *NeurIPS*, 2020.
[147] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
[148] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised gans via auxiliary rotation loss," in *CVPR*, 2019.
[149] J. Wang, W. Zhou, G.-J. Qi, Z. Fu, Q. Tian, and H. Li, "Transformation gan for unsupervised image synthesis and representation learning," in *CVPR*, 2020.
[150] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *ICLR*, 2019.
[151] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," in *NeurIPS*, 2019.
[152] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *ICCV*, 2015.
[153] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, "Tracking emerges by colorizing videos," in *ECCV*, 2018.
[154] I. Aganj, M. G. Harisinghani, R. Weissleder, and B. Fischl, "Unsupervised medical image segmentation based on the local center of mass," *Nature*, 2018.
[155] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pre-training," in *CVPR*, 2019.
[156] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He, "A large-scale study on unsupervised spatiotemporal representation learning," in *CVPR*, 2021.
[157] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE TPAMI*, 2016.
[158] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.
[159] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *CVPR*, 2017.
[160] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.
[161] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022.
[162] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould, "Visual permutation learning," *IEEE TPAMI*, 2018.
[163] T. Nathan Mundhenk, D. Ho, and B. Y. Chen, "Improvements to context based self-supervised learning," in *CVPR*, 2018.
[164] C. Wei, L. Xie, X. Ren, Y. Xia, C. Su, J. Liu, Q. Tian, and A. L. Yuille, "Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning," in *CVPR*, 2019.
[165] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and benchmarking self-supervised visual representation learning," in *ICCV*, 2019.
[166] G. Goh, N. Cammarata, C. Voss, S. Carter, M. Petrov, L. Schubert, A. Radford, and C. Olah, "Multimodal neurons in artificial neural networks," *Distill*, 2021.
[167] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *CVPR*, 2019.
[168] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *ICLR*, 2019.
[169] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *NeurIPS*, 2019.
[170] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, "On mutual information maximization for representation learning," in *ICLR*, 2019.
[171] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning," in *NeurIPS*, 2020.
[172] J. Huang, Q. Dong, S. Gong, and X. Zhu, "Unsupervised deep learning by neighbourhood discovery," in *ICML*, 2019.
[173] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool, "Learning to classify images without labels," in *ECCV*, 2020.
[174] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, "Unsupervised pre-training of image features on non-curated data," in *ICCV*, 2019.
[175] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self-supervised learning of geometrically stable features through probabilistic introspection," in *CVPR*, 2018.
[176] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *CVPR*, 2019.
[177] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.
[178] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010.
[179] M. Federici, A. Dutta, P. Forré, N. Kushman, and Z. Akata, "Learning robust representations via multi-view information bottleneck," in *ICLR*, 2020.
[180] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
[181] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *ICML*, 2021.
[182] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016.
[183] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama, "Learning discrete representations via information maximizing self-augmented training," in *ICML*, 2017.
[184] C. Zhuang, A. L. Zhai, and D. Yamins, "Local aggregation for unsupervised learning of visual embeddings," in *CVPR*, 2019.
[185] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram, and D. Mahajan, "Clusterfit: Improving generalization of visual representations," in *CVPR*, 2020.
[186] X. Wang, Z. Liu, and S. X. Yu, "Unsupervised feature learning by cross-level discrimination between instances and groups," in *CVPR*, 2021.
[187] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Learning representations by predicting bags of visual words," in *CVPR*, 2020.
[188] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *CSUR*, 1999.
[189] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural networks: Tricks of the trade*, 2012.
[190] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, 2007.
[191] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *CVPR*, 2017.
[192] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *IJCAI*, 2017.
[193] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *ICML*, 2017.
[194] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *PR*, 1978.
[195] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *NeurIPS*, 2013.
[196] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *AAAI*, 2020.

[197] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, 2006.

[198] A. Gabbay and Y. Hoshen, "Demystifying inter-class disentanglement," in *ICLR*, 2020.

[199] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *ICLR*, 2017.

[200] G.-J. Qi, L. Zhang, C. W. Chen, and Q. Tian, "Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations," in *ICCV*, 2019.

[201] L. Zhang, G.-J. Qi, L. Wang, and J. Luo, "Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data," in *CVPR*, 2019.

[202] X. Chen, Y. Yuan, G. Zeng, and J. Wang, "Semi-supervised semantic segmentation with cross pseudo supervision," in *CVPR*, 2021.

[203] J. Jeong, S. Lee, J. Kim, and N. Kwak, "Consistency-based semi-supervised learning for object detection," in *NeurIPS*, 2019.

[204] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.

[205] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, "Universal domain adaptation through self supervision," in *NeurIPS*, 2020.

[206] L. Yang, S. Chen, and A. Yao, "Semihand: Semi-supervised hand pose estimation with consistency," in *ICCV*, 2021.

[207] T. Kim, J. Choi, S. Choi, D. Jung, and C. Kim, "Just a few points are all you need for multi-view stereo: A novel semi-supervised learning method for multi-view stereo," in *ICCV*, 2021.

[208] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, "Self-supervised learning by cross-modal audio-video clustering," in *NeurIPS*, 2020.

[209] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[210] X. Huo, L. Xie, J. He, Z. Yang, W. Zhou, H. Li, and Q. Tian, "Atso: Asynchronous teacher-student optimization for semi-supervised image segmentation," in *CVPR*, 2021.

[211] H. Wu, G. Chen, Z. Wen, and J. Qin, "Collaborative and adversarial learning of focused and dispersive representations for semi-supervised polyp segmentation," in *ICCV*, 2021.

[212] H. Huang, L. Lin, Y. Zhang, Y. Xu, J. Zheng, X. Mao, X. Qian, Z. Peng, J. Zhou, Y.-W. Chen *et al.*, "Graph-bas3net: Boundary-aware semi-supervised segmentation network with bilateral graph convolution," in *ICCV*, 2021.

[213] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[214] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *CVPR*, 2018.

[215] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *ICCV*, 2019.

[216] G. French, T. Aila, S. Laine, M. Mackiewicz, and G. Finlayson, "Semi-supervised semantic segmentation needs strong, high-dimensional perturbations," in *BMVC*, 2020.

[217] Y. Ouali, C. Hudelot, and M. Tami, "Semi-supervised semantic segmentation with cross-consistency training," in *CVPR*, 2020.

[218] Z. Ke, D. Qiu, K. Li, Q. Yan, and R. W. Lau, "Guided collaborative training for pixel-wise semi-supervised learning," in *ECCV*, 2020.

[219] H. Hu, F. Wei, H. Hu, Q. Ye, J. Cui, and L. Wang, "Semi-supervised semantic segmentation via adaptive equalization learning," in *NeurIPS*, 2021.

[220] R. Mendel, L. A. De Souza, D. Rauber, J. P. Papa, and C. Palm, "Semi-supervised segmentation based on error-correcting supervision," in *ECCV*, 2020.

[221] Y. Zou, Z. Zhang, H. Zhang, C.-L. Li, X. Bian, J.-B. Huang, and T. Pfister, "Pseudoseg: Designing pseudo labels for semantic segmentation," in *ICLR*, 2021.

[222] M. S. Ibrahim, A. Vahdat, M. Ranjbar, and W. G. Macready, "Semi-supervised semantic image segmentation with self-correcting networks," in *CVPR*, 2020.

[223] R. He, J. Yang, and X. Qi, "Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation," in *ICCV*, 2021.

[224] J. Yuan, Y. Liu, C. Shen, Z. Wang, and H. Li, "A simple baseline for semi-supervised semantic segmentation with strong data augmentation," in *ICCV*, 2021.

[225] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *ICCV*, 2017.

[226] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, "Adversarial learning for semi-supervised semantic segmentation," in *BMVC*, 2018.

[227] S. Mittal, M. Tatarchenko, and T. Brox, "Semi-supervised semantic segmentation with high-and low-level consistency," *IEEE TPAMI*, 2021.

[228] X. Lai, Z. Tian, L. Jiang, S. Liu, H. Zhao, L. Wang, and J. Jia, "Semi-supervised semantic segmentation with directional context-aware consistency," in *CVPR*, 2021.

[229] I. Alonso, A. Sabater, D. Ferstl, L. Montesano, and A. C. Murillo, "Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank," in *ICCV*, 2021.

[230] Y. Zhong, B. Yuan, H. Wu, Z. Yuan, J. Peng, and Y.-X. Wang, "Pixel contrastive-consistent semi-supervised semantic segmentation," in *ICCV*, 2021.

[231] Y. Zhou, H. Xu, W. Zhang, B. Gao, and P.-A. Heng, "C3-semiseg: Contrastive semi-supervised segmentation via cross-set learning and dynamic class-balancing," in *ICCV*, 2021.

[232] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019.

[233] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[234] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *CVPR*, 2017.

[235] K. Qian, S. Zhu, X. Zhang, and L. E. Li, "Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals," in *CVPR*, 2021.

[236] H. Su, S. Gong, and X. Zhu, "Multi-perspective cross-class domain adaptation for open logo detection," *CVIU*, 2021.

[237] W. Feng, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Semantic-aware video text detection," in *CVPR*, 2021.

[238] O. Russakovsky, L.-J. Li, and L. Fei-Fei, "Best of both worlds: human-machine collaboration for object annotation," in *CVPR*, 2015.

[239] J. Jeong, V. Verma, M. Hyun, J. Kannala, and N. Kwak, "Interpolation-based semi-supervised learning for object detection," in *CVPR*, 2021.

[240] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, "Unbiased teacher for semi-supervised object detection," in *ICLR*, 2021.

[241] Q. Zhou, C. Yu, Z. Wang, Q. Qian, and H. Li, "Instant-teaching: An end-to-end semi-supervised object detection framework," in *CVPR*, 2021.

[242] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, "End-to-end semi-supervised object detection with soft teacher," in *ICCV*, 2021.

[243] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister, "A simple semi-supervised learning framework for object detection," *arXiv preprint arXiv:2005.04757*, 2020.

[244] Q. Yang, X. Wei, B. Wang, X.-S. Hua, and L. Zhang, "Interactive self-training with mean teachers for semi-supervised object detection," in *CVPR*, 2021.

[245] Z. Wang, Y. Li, Y. Guo, L. Fang, and S. Wang, "Data-uncertainty guided multi-phase learning for semi-supervised object detection," in *CVPR*, 2021.

[246] Z. Wang, Y. Li, Y. Guo, and S. Wang, "Combating noise: Semi-supervised learning by region uncertainty quantification," in *NeurIPS*, 2021.

[247] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010.

[248] Y. Wu, D. Inkpen, and A. El-Roby, "Dual mixup regularized learning for adversarial domain adaptation," in *ECCV*, 2020.

[249] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," in *ICLR*, 2018.

[250] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *ICCV*, 2019.

[251] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, "Autodial: Automatic domain alignment layers," in *ICCV*, 2017.

[252] P. Morerio, J. Cavazza, and V. Murino, "Minimal-entropy correlation alignment for unsupervised deep domain adaptation," in *ICLR*, 2018.

[253] L. Hu, M. Kan, S. Shan, and X. Chen, "Unsupervised domain adaptation with hierarchical gradient synchronization," in *CVPR*, 2020.

[254] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *ICML*, 2017.

[255] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *CVPR*, 2017.

[256] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Nam Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *CVPR*, 2018.

[257] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *ICML*, 2018.

[258] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo, "From source to target and back: symmetric bi-directional adaptive gan," in *CVPR*, 2018.

[259] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *CVPR*, 2018.

[260] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Crdoco: Pixel-level domain transfer with cross-domain consistency," in *CVPR*, 2019.

[261] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *ECCV*, 2016.

[262] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training." in *CVPR*, 2017.

[263] J. Xu, L. Xiao, and A. M. López, "Self-supervised domain adaptation for computer vision tasks," *IEEE Access*, 2019.

[264] S. Bucci, A. D'Innocente, Y. Liao, F. M. Carlucci, B. Caputo, and T. Tommasi, "Self-supervised learning across domains," *IEEE TPAMI*, 2020.

[265] R. Wang, Z. Wu, Z. Weng, J. Chen, G.-J. Qi, and Y.-G. Jiang, "Cross-domain contrastive learning for unsupervised domain adaptation," *IEEE Transactions on Multimedia*, 2022.

[266] P. Su, S. Tang, P. Gao, D. Qiu, N. Zhao, and X. Wang, "Gradient regularized contrastive learning for continual domain adaptation," in *AAAI*, 2021.

[267] Q. Yu, D. Ikami, G. Irie, and K. Aizawa, "Multi-task curriculum framework for open-set semi-supervised learning," in *ECCV*, 2020.

[268] M. Augustin and M. Hein, "Out-distribution aware self-training in an open world setting," *arXiv preprint arXiv:2012.12372*, 2020.

[269] J. Huang, C. Fang, W. Chen, Z. Chai, X. Wei, P. Wei, L. Lin, and G. Li, "Trash to treasure: Harvesting ood data with cross-modal matching for open-set semi-supervised learning," in *ICCV*, 2021.

[270] K. Saito, D. Kim, and K. Saenko, "Openmatch: Open-set consistency regularization for semi-supervised learning with outliers," in *NeurIPS*, 2021.

[271] Z. Huang, C. Xue, B. Han, J. Yang, and C. Gong, "Universal semi-supervised learning," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[272] K. Cao, M. Brbic, and J. Leskovec, "Open-world semi-supervised learning," in *ICLR*, 2022.

[273] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *ICLR*, 2017.

[274] Z. Zhong, L. Zhu, Z. Luo, S. Li, Y. Yang, and N. Sebe, "Openmix: Reviving known knowledge for discovering novel visual categories in an open world," in *CVPR*, 2021.

[275] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *ICLR*, 2019.

[276] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *NeurIPS*, 2018.

[277] M. Hein, M. Andriushchenko, and J. Bitterwolf, "Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem," in *CVPR*, 2019.

[278] J. Kim, Y. Hur, S. Park, E. Yang, S. J. Hwang, and J. Shin, "Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning," in *NeurIPS*, 2020.

[279] H. Lee, S. Shin, and H. Kim, "Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning," in *NeurIPS*, 2021.

[280] K. Han, A. Vedaldi, and A. Zisserman, "Learning to discover novel visual categories via deep transfer clustering," in *ICCV*, 2019.

[281] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman, "Automatically discovering and learning new visual categories with ranking statistics," in *ICLR*, 2020.

[282] K. Lee, K. Lee, J. Shin, and H. Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," in *ICCV*, 2019.

[283] D. Rao, F. Visin, A. A. Rusu, R. Pascanu, Y. W. Teh, and R. Hadsell, "Continual unsupervised representation learning," in *NeurIPS*, 2019.

[284] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE TPAMI*, 2021.

[285] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, 1989.

[286] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *CVPR*, 2017.

[287] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *PNAS*, 2017.

[288] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *ECCV*, 2018.

[289] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *ECCV*, 2018.

[290] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *WACV*, 2020.

[291] Y. Li, Y. Wang, Q. Liu, C. Bi, X. Jiang, and S. Sun, "Incremental semi-supervised learning on streaming data," *PR*, 2019.

[292] J. Smith, S. Baer, Z. Kira, and C. Dovrolis, "Unsupervised continual learning and self-taught associative memory hierarchies," in *ICLRW*, 2019.

[293] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *ICML*, 2020.

[294] T. Varsavsky, M. Orbes-Arteaga, C. H. Sudre, M. S. Graham, P. Nachev, and M. J. Cardoso, "Test-time unsupervised domain adaptation," in *MICCAI*, 2020.

[295] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimiaztion," in *ICLR*, 2021.

[296] J. Hoffman, T. Darrell, and K. Saenko, "Continuous manifold based adaptation for evolving visual domains," in *CVPR*, 2014.

[297] A. Bobu, E. Tzeng, J. Hoffman, and T. Darrell, "Adapting to continuously shifting domains," in *ICLRW*, 2018.

[298] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *IJCNN*, 2020.

[299] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS*, 2019.

[300] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," in *ACL*, 2019.

[301] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," in *ICLR*, 2019.

[302] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *ECCV*, 2020.

[303] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang, and M. Zhou, "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training." in *AAAI*, 2020.

[304] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[305] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *ACL*, 2019.

[306] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *ICCV*, 2019.
[307] J. C. Stroud, D. A. Ross, C. Sun, J. Deng, R. Sukthankar, and C. Schmid, "Learning video representations from textual web supervision," *arXiv preprint arXiv:2007.14937*, 2020.
[308] J.-B. Alayrac, A. Recasens, R. Schneider, R. Arandjelović, J. Ramapuram, J. De Fauw, L. Smaira, S. Dieleman, and A. Zisserman, "Self-supervised multimodal versatile networks," in *NeurIPS*, 2020.
[309] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *CVPR*, 2020.
[310] A. Rouditchenko, A. Boggust, D. Harwath, D. Joshi, S. Thomas, K. Audhkhasi, R. Feris, B. Kingsbury, M. Picheny, A. Torralba *et al.*, "Avlnet: Learning audio-visual language representations from instructional videos," *arXiv preprint arXiv:2006.09199*, 2020.
[311] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *ICCV*, 2019.
[312] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.
[313] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *ICML*, 2021.
[314] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *CVPR*, 2017.
[315] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *ECCV*, 2018.
[316] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *NeurIPS*, 2018.
[317] P. Morgado, N. Vasconcelos, and I. Misra, "Audio-visual instance discrimination with cross-modal agreement," in *CVPR*, 2021.
[318] M. Patrick, Y. M. Asano, R. Fong, J. F. Henriques, G. Zweig, and A. Vedaldi, "Multi-modal self-supervision from generalized data transformations," *arXiv preprint arXiv:2003.04298*, 2020.
[319] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning," in *ECCV*, 2016.
[320] P. Hu, H. Zhu, X. Peng, and J. Lin, "Semi-supervised multi-modal learning with balanced spectral decomposition," in *AAAI*, 2020.
[321] S. Li, B. Xie, J. Wu, Y. Zhao, C. H. Liu, and Z. Ding, "Simultaneous semantic alignment network for heterogeneous domain adaptation," in *ACM MM*, 2020.
[322] Y. Chen, Y. Xian, A. S. Koepke, Y. Shan, and Z. Akata, "Distilling audio-visual knowledge by compositional contrastive learning," in *CVPR*, 2021.

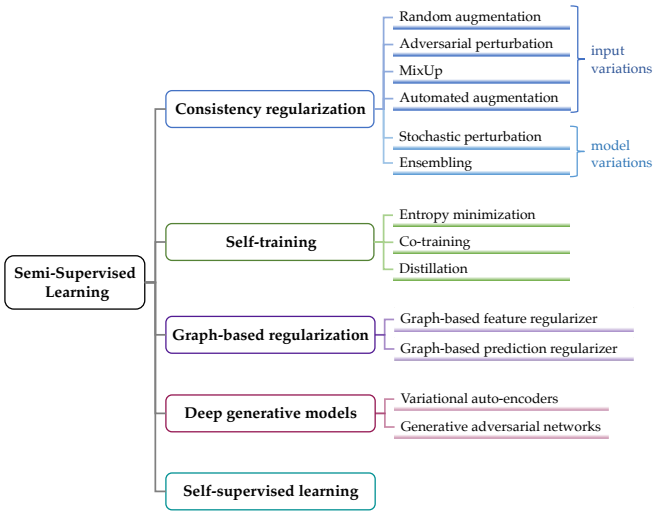


Fig. A: A summary taxonomy of semi-supervised learning.

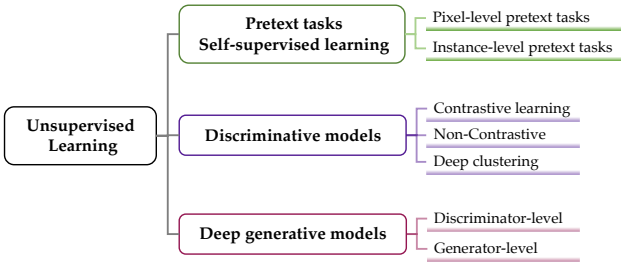


Fig. B: A summary taxonomy of unsupervised learning.

SUPPLEMENTARY MATERIALS

TABLE A: The problem setups of SSL and UL. $\mathcal{D}_L, \mathcal{D}_U$ denote labeled and unlabeled data. "–": no training data.

Problem Setups	Training Stage I	Training Stage II
Semi-supervised learning (SSL)	$\mathcal{D}_L \cup \mathcal{D}_U$	–
Unsupervised learning (UL)	$\mathcal{D}_U$	$\mathcal{D}_L$

TABLE B: An overview of the SSL and UL algorithms used in three popular visual recognition tasks.

SSL & UL methods	**applied tasks**
consistency regularization (SSL, §2.2.1) self-training (SSL, §2.2.2)	semantic segmentation object detection domain adaptation
discriminative models (UL, §3.2.2)	semantic segmentation domain adaptation
pretext tasks (UL, §3.2.1) deep generative models (SSL/UL, §2.2.4, §3.2.3)	domain adaptation