Self-Supervised Learning: Generative or Contrastive

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang, *Fellow, IEEE*

Abstract—Deep supervised learning has achieved great success in the last decade. However, its defects of heavy dependence on manual labels and vulnerability to attacks have driven people to find other paradigms. As an alternative, self-supervised learning (SSL) attracts many researchers for its soaring performance on representation learning in the last several years. Self-supervised representation learning leverages input data itself as supervision and benefits almost all types of downstream tasks. In this survey, we take a look into new self-supervised learning methods for representation in computer vision, natural language processing, and graph learning. We comprehensively review the existing empirical methods and summarize them into three main categories according to their objectives: generative, contrastive, and generative-contrastive (adversarial). We further collect related theoretical analysis on self-supervised learning to provide deeper thoughts on why self-supervised learning works. Finally, we briefly discuss open problems and future directions for self-supervised learning. An outline slide for the survey is provided[1].

Index Terms—Self-supervised learning, generative model, contrastive learning, deep learning

✦

1 INTRODUCTION

DEEP neural networks [71] have shown outstanding performance on various machine learning tasks, especially on supervised learning in computer vision (image classification [26], [48], [54], semantic segmentation [39], [76]), natural language processing (pre-trained language models [27], [68], [75], [138], sentiment analysis [74], question answering [5], [29], [100], [139] etc.) and graph learning (node classification [53], [64], [95], [126], graph classification [7], [110], [146] etc.). Generally, the supervised learning is trained on a specific task with a large labeled dataset, which is randomly divided for training, validation and test.

However, supervised learning is meeting its bottleneck. It relies heavily on expensive manual labeling and suffers from generalization error, spurious correlations, and adversarial attacks. We expect the neural networks to learn more

with fewer labels, fewer samples, and fewer trials. As a promising alternative, self-supervised learning has drawn massive attention for its data efficiency and generalization ability, and many state-of-the-art models have been following this paradigm. This survey will take a comprehensive look at the recent developing self-supervised learning models and discuss their theoretical soundness, including frameworks such as Pre-trained Language Models (PTM), Generative Adversarial Networks (GAN), autoencoders and their extensions, Deep Infomax, and Contrastive Coding. An outline slide is also provided.[1]

The term "self-supervised learning" is first introduced in robotics, where training data is automatically labeled by leveraging the relations between different input sensor signals. Afterwards, machine learning community further develops the idea. In the invited speech on AAAI 2020, The Turing award winner Yann LeCun described self-supervised learning as "the machine predicts any parts of its input for any observed part".[2] Combining self-supervised learning's traditional definition and LeCun's definition, we can further summarize its features as:

- Obtain "labels" from the data itself by using a "semi-automatic" process.
- Predict part of the data from other parts.

Specifically, the "other part" could be incomplete, transformed, distorted, or corrupted (i.e., data augmentation technique). In other words, the machine learns to 'recover' whole, or parts of, or merely some features of its original input.

People are often confused by the concepts of unsupervised learning and self-supervised learning. Self-supervised

- Xiao Liu, Fanjin Zhang, and Zhenyu Hou are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100190, China. E-mail: {liuxiao17, zfj17, hzy17}@mails.tsinghua.edu.cn.
- Li Mian is with the Beijing Institute of Technonlogy, Beijing 100811, China. E-mail: 1120161659@bit.edu.cn.
- Zhaoyu Wang is with Anhui University, Anhui 230093, China. E-mail: wzy950507@163.com.
- Jing Zhang is with the Renmin University of China, Beijing 100872, China. E-mail: zhang-jing@ruc.edu.cn.
- Jie Tang is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100190, China, and also with the Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing 100084, China. E-mail: jietang@tsinghua.edu.cn.

1. Slides at https://www.aminer.cn/pub/5ee8986f91e011e66831c59b/
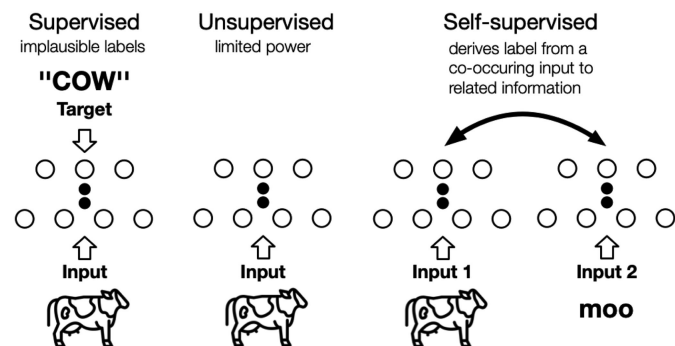2. https://aaai.org/Conferences/AAAI-20/invited-speakers/

Fig. 1. An illustration to distinguish the supervised, unsupervised and self-supervised learning framework. In self-supervised learning, the "related information" could be another modality, parts of inputs, or another form of the inputs. Repainted from [25].
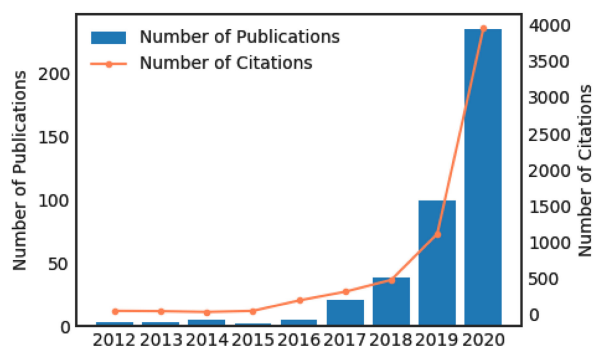

Fig. 2. Number of publications and citations on self-supervised learning during 2012-2020, from Microsoft Academic [108], [144]. Self-supervised learning is drawing tremendous attention in recent years.

learning can be viewed as a branch of unsupervised learning since there is no manual label involved. However, narrowly speaking, unsupervised learning concentrates on detecting specific data patterns, such as clustering, community discovery, or anomaly detection, while self-supervised learning aims at recovering, which is still in the paradigm of supervised settings. Fig. 1 provides a vivid explanation of the differences between them.

There exist several comprehensive reviews related to Pre-trained Language Models [96], Generative Adversarial Networks [130], autoencoders, and contrastive learning for visual representation [57]. However, none of them concentrates on the inspiring idea of self-supervised learning itself. In this work, we collect studies from natural language processing, computer vision, and graph learning in recent years to present an up-to-date and comprehensive retrospective on the frontier of self-supervised learning. To sum up, our contributions are:

- We provide a detailed and up-to-date review of self-supervised learning for representation. We introduce the background knowledge, models with variants, and important frameworks. One can easily grasp the frontier ideas of self-supervised learning.
- We categorize self-supervised learning models into generative, contrastive, and generative-contrastive (adversarial), with particular genres inner each one. We demonstrate the pros and cons of each category.
- We identify several open problems in this field, analyze the limitations and boundaries, and discuss the future direction for self-supervised representation learning.

We organize the survey as follows. In Section 2, we introduce the motivation of self-supervised learning. We also present our categorization of self-supervised learning and a conceptual comparison between them. From Sections 3, 4, and 5, we will introduce the empirical self-supervised learning methods utilizing generative, contrastive and generative-contrastive objectives. Finally, in Sections 6 and 7, we discuss the open problems, future directions and our conclusions.

2 MOTIVATION OF SELF-SUPERVISED LEARNING

It is universally acknowledged that deep learning algorithms are data-hungry. Compared to traditional feature-based methods, deep learning usually follows the so-called "end-to-end" fashion (raw-data in, prediction out). It makes very few prior assumptions, which leads to over-fitting and biases in scenarios with little supervised data. Literature has shown that simple multi-layer perceptrons have a very poor generalization ability (always assume a linear relationship for out-of-distribution (OOD) samples) [135], which results in over-confident (and wrong) predictions.

To conquer the fundamental OOD and generalization problem, while numerous works focus on designing new architectures for neural networks, another simple yet effective solution is to enlarge the training dataset to make as many samples "in-distribution". However, the fact is, despite massive available unlabeled web data in this big data era, high-quality data with human labeling could be costly. For example, Scale.ai,[3] a data labeling company, charges $6.4 per image for the image segmentation labeling. An image segmentation dataset containing 10k+ high-quality samples could cost up to a million-dollar.

The most crucial point for self-supervised learning's success is that it figures out a way to leverage the tremendous amounts of unlabeled data that becomes available in the big data era. It a time for deep learning algorithms to get rid of human supervision and turn back to data's *self-supervision*. The intuition of self-supervised learning is to leverage the data's inherent co-occurrence relationships as the self-supervision, which could be versatile. For example, in the incomplete sentence "I like ___ apples", a well-trained language model would predict "eating" for the blank (i.e., the famous Cloze Test [117]) because it frequently co-occurs with the context in the corpora. We can summarize the mainstream self-supervision into three general categories (see Fig. 3) and detailed subsidiaries:

- Generative: train an encoder to encode input x into an explicit vector z and a decoder to reconstruct x from z (e.g., the cloze test, graph generation)
- Contrastive: train an encoder to encode input x into an explicit vector z to measure similarity (e.g., mutual information maximization, instance discrimination)
- Generative-Contrastive (Adversarial): train an encoder-decoder to generate fake samples and a discriminator to distinguish them from real samples (e.g., GAN)
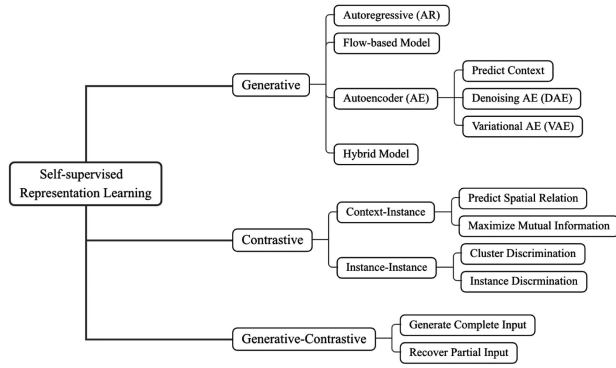
3. https://scale.com/pricing

Fig. 3. Categorization of Self-supervised learning (SSL): Generative, Contrastive and Generative-Contrastive (Adversarial).

Fig. 4. Conceptual comparison between Generative, Contrastive, and Generative-Contrastive methods.

Their main difference lies in model architectures and objectives. A detailed conceptual comparison is shown in Fig. 4. Their architectures can be unified into two general components: the generator and the discriminator, and the generator can be further decomposed into an encoder and a decoder. Different things are:

1) For latent distribution z: in generative and contrastive methods, z is explicit and is often leveraged by downstream tasks; while in GAN, z is implicitly modeled.
2) For discriminator: the generative method does not have a discriminator while GAN and contrastive have. Contrastive discriminator has comparatively fewer parameters (e.g., a multi-layer perceptron with 2-3 layers) than GAN (e.g., a standard ResNet [48]).
3) For objectives: the generative methods use a reconstruction loss, the contrastive ones use a contrastive similarity metric (e.g., InfoNCE), and the generative-contrastive ones leverage distributional divergence as the loss (e.g., JS-divergence, Wasserstein Distance).

A properly designed training objective related to downstream tasks could turn our randomly initialized models into excellent pre-trained feature extractors. For example, contrastive learning is found to be useful for almost all visual classification tasks. This is probably because the contrastive object is modeling the class-invariance between different image instances. The contrastive loss makes images containing the same object class more similar. It makes those containing different classes less similar, essentially accords with the downstream image classification, object detection, and other classification-based tasks. The art of self-supervised learning primarily lies in defining proper objectives for unlabeled data.

3 GENERATIVE SELF-SUPERVISED LEARNING

This section will introduce important self-supervised learning methods based on generative models, including auto-regressive (AR) models, flow-based models, auto-encoding (AE) models, and hybrid generative models.

3.1 Auto-Regressive (AR) Model

Auto-regressive (AR) models can be viewed as "Bayes net structure" (directed graph model). The joint distribution

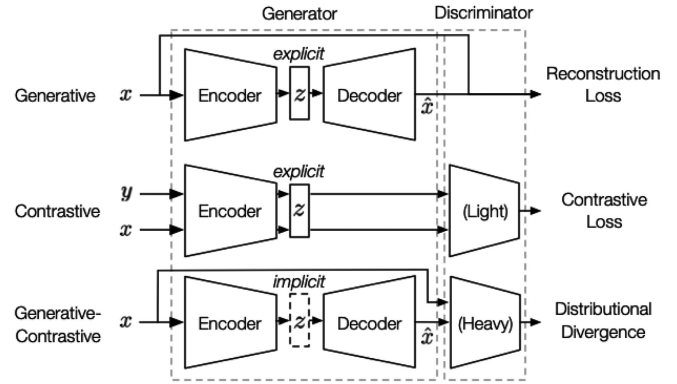can be factorized as a product of conditionals

$$\max_{\theta} p_{\theta}(\mathbf{x}) = \sum_{t=1}^{T} \log p_{\theta}(x_t | \mathbf{x}_{1:t-1}), \quad (1)$$

where the probability of each variable is dependent on the previous variables.

In NLP, the objective of auto-regressive language modeling is usually maximizing the likelihood under the forward autoregressive factorization [138]. GPT [98] and GPT-2 [99] use Transformer decoder architecture [125] for language model. Different from GPT, GPT-2 removes the fine-tuning processes of different tasks. To learn unified representations that generalize across different tasks, GPT-2 models $p(output|input, task)$, which means given different tasks, the same inputs can have different outputs.

The auto-regressive models have also been employed in computer vision, such as PixelRNN [124] and PixelCNN [122]. The general idea is to use auto-regressive methods to model images pixel by pixel. For example, the lower (right) pixels are generated by conditioning on the upper (left) pixels. The pixel distributions of PixelRNN and PixelCNN are modeled by RNN and CNN, respectively. For 2D images, auto-regressive models can only factorize probabilities according to specific directions (such as right and down). Therefore, masked filters are employed in CNN architecture. Furthermore, two convolutional networks are combined to remove the blind spot in images. Based on PixelCNN, WaveNet [121] – a generative model for raw audio was proposed. To deal with long-range temporal dependencies, the authors develop dilated causal convolutions to improve the receptive field. Moreover, Gated Residual blocks and skip connections are employed to empower better expressivity.

The auto-regressive models can also be applied to graph domain problems, such as graph generation. You et al. [141] propose GraphRNN to generate realistic graphs with deep auto-regressive models. They decompose the graph generation process into a sequence generation of nodes and edges conditioned on the graph generated so far. The objective of GraphRNN is defined as the likelihood of the observed graph generation sequences. GraphRNN can be viewed as a hierarchical model, where a graph-level RNN maintains the state of the graph and generates new nodes, while an edge-level RNN generates new edges based on the current graph

TABLE 1
An Overview of Recent Self-Supervised Representation Learning

Model	FOS	Type	Generator	Self-supervision	Pretext Task	Hard NS	Hard PS	NS strategy
GPT/GPT-2 [98], [99]	NLP	G	AR	Following words	Next word prediction	-	-	-
PixelCNN [122], [124]	CV	G	AR	Following pixels	Next pixel prediction	-	-	-
NICE [30]	CV	G	Flow	Whole image	Image reconstruction	-	-	-
RealNVP [31]	CV	G	based			-	-	-
Glow [62]	CV	G				-	-	-
word2vec [79], [80]	NLP	G	AE	Context words	CBOW & SkipGram	×	×	End-to-end
FastText [10]	NLP	G	AE		CBOW	×	×	End-to-end
DeepWalk-based [43], [92], [115]	Graph	G	AE	Graph edges	Link prediction	×	×	End-to-end
VGAE [65]	Graph	G	AE			×	×	End-to-end
BERT [27]	NLP	G	AE	Masked words Sentence topic	Masked language model, Next senetence prediction	-	-	-
SpanBERT [58]	NLP	G	AE	Masked words	Masked language model	-	-	-
ALBERT [68]	NLP	G	AE	Masked words Sentence order	Masked language model, Sentence order prediction	-	-	-
ERNIE [114], [149]	NLP	G	AE	Masked words Sentence topic	Masked language model, Next senetence prediction	-	-	-
GPT-GNN [52]	Graph	G	AE	Attribute & Edge	Masked graph generation	-	-	-
VQ-VAE 2 [101]	CV	G	AE	Whole image	Image reconstruction	-	-	-
XLNet [138]	NLP	G	AE+AR	Masked words	Permutation language model	-	-	-
GraphAF [107]	Graph	G	Flow+AR	Attribute & Edge	Sequential graph generation	-	-	-
RelativePosition [32]	CV	C	-	Spatial relations (Context-Instance)	Relative postion prediction	-	-	-
CDJP [61]	CV	C	-		Jigsaw + Inpainting + Colorization	×	×	End-to-end
PIRL [81]	CV	C	-		Jigsaw	×	✓	Memory bank
RotNet [38]	CV	C	-		Rotation Prediction	-	-	-
Deep InfoMax [49]	CV	C	-	Belonging (Context-Instance)	MI Maximization	×	×	End-to-end
AMDIM [6]	CV	C	-			×	✓	End-to-end
CPC [88]	CV	C	-			×	×	End-to-end
InfoWord [66]	NLP	C	-			×	×	End-to-end
DGI [127]	Graph	C	-			✓	×	End-to-end
InfoGraph [110]	Graph	C	-			×	×	End-to-end (batch-wise)
CMC-Graph [46]	Graph	C	-			×	✓	End-to-end
S^{2}GRL [90]	Graph	C	-			×	×	End-to-end
Pre-trained GNN [51]	Graph	C	-	Belonging Node attributes	MI maximization, Masked attribute prediction	×	×	End-to-end
DeepCluster [13]	CV	C	-	Similarity (Instance-Instance)	Cluster discrimination	-	-	-
Local Aggregation [152]	CV	C	-			-	-	-
ClusterFit [136]	CV	C	-			-	-	-
SwAV [14]	CV	C	-			-	✓	End-to-end
SEER [41]	CV	C	-			-	✓	End-to-end
M3S [113]	Graph	C	-			-	-	-
InstDisc [132]	CV	C	-	Identity (Instance-Instance)	Instance discrimination	×	×	Memory bank
CMC [118]	CV	C	-			×	✓	End-to-end
MoCo [47]	CV	C	-			×	×	Momentum
MoCo v2 [18]	CV	C	-			×	✓	Momentum
SimCLR [15]	CV	C	-			×	✓	End-to-end
InfoMin [119]	CV	C	-			×	✓	End-to-end
BYOL [42]	CV	C	-			no NS	✓	End-to-end
ReLIC [82]	CV	C	-			×	✓	End-to-end
SimSiam [19]	CV	C	-			no NS	✓	End-to-end
SimCLR v2 (semi) [16]	CV	C	-			×	✓	End-to-end
GCC [94]	Graph	C	-			×	✓	Momentum
GraphCL [142]	Graph	C	-			×	✓	End-to-end
GAN [40]	CV	G-C	AE	Whole image	Image reconstruction	-	-	-
Adversarial AE [77]	CV	G-C	AE			-	-	-
BiGAN/ALI [33], [36]	CV	G-C	AE			-	-	-
BigBiGAN [34]	CV	G-C	AE			-	-	-
Colorization [69]	CV	G-C	AE	Image color	Colorization	-	-	-
Inpainting [89]	CV	G-C	AE	Parts of images	Inpainting	-	-	-
Super-resolution [72]	CV	G-C	AE	Details of images	Super-resolution	-	-	-
ELECTRA [21]	NLP	G-C	AE	Masked words	Replaced token detection	✓	×	End-to-end

TABLE 1
(*Continued*)

Model	FOS	Type	Generator	Self-supervision	Pretext Task	Hard NS	Hard PS	NS strategy
WKLM [134]	NLP	G-C	AE	Masked entities	Replaced entity detection	✓	×	End-to-end
ANE [23]	Graph	G-C	AE	Graph edges	Link prediction	-	-	-
GraphGAN [128]	Graph	G-C	AE			-	-	-
GraphSGAN [28]	Graph	G-C	AE	Graph nodes	Node classification	-	-	-

For acronyms used, "FOS" refers to fields of study; "NS" refers to negative samples; "PS" refers to positive samples; "MI" refers to mutual information. For alphabets in "Type": G Generative; C Contrastive; G-C Generative-Contrastive (Adversarial). For symbols in "Hard NS" and "Hard PS", "-" means not applicable, "×" means not adopted, "✓"' means adopted; "no NS" particularly means not using negative samples in instance-instance contrast.

state. After that, MRNN [93] and GCPN [140] are proposed as auto-regressive approaches. MRNN and GCPN both use a reinforcement learning framework to generate molecule graphs through optimizing domain-specific rewards. However, MRNN mainly uses RNN-based networks for state representations, but GCPN employs GCN-based encoder networks.

The advantage of auto-regressive models is that they can model the context dependency well. However, one shortcoming of the AR model is that the token at each position can only access its context from one direction.

3.2 Flow-Based Model

The goal of flow-based models is to estimate complex high-dimensional densities $p(x)$ from data. Intuitively, directly formalizing the densities is difficult. To obtain a complicated densities, we hope to generate it "step by step" by stacking a series of transforming functions that describing different data characteristics respectively. Generally, flow-based models first define a latent variable z which follows a known distribution $p_Z(z)$. Then define $z = f_\theta(x)$, where f_θ is an invertible and differentiable function. The goal is to learn the transformation between x and z so that the density of x can be depicted. According to the integral rule, $p_\theta(x)dx = p(z)dz$. Therefore, the densities of x and z satisfies

$$p_\theta(x) = p(f_\theta(x)) \left| \frac{\partial f_\theta(x)}{\partial x} \right|, \tag{2}$$

and the objective is to maximize the likelihood

$$\max_\theta \sum_i \log p_\theta(x^{(i)}) = \max_\theta \sum_i \log p_Z(f_\theta(x^{(i)}))$$
$$+ \log \left| \frac{\partial f_\theta}{\partial x}(x^{(i)}) \right|. \tag{3}$$

The advantage of flow-based models is that the mapping between x and z is invertible. However, it also requires that x and z must have the same dimension. f_θ needs to be carefully designed since it should be invertible and the Jacobian determinant in Eq. (2) should also be calculated easily. NICE [30] and RealNVP [31] design affine coupling layer to parameterize f_θ. The core idea is to split x into two blocks (x_1, x_2) and apply a transformation from (x_1, x_2) to (z_1, z_2) in an auto-regressive manner, that is $z_1 = x_1$ and $z_2 = x_2 + m(x_1)$. More recently, Glow [62] was proposed and it introduces invertible 1×1 convolutions and simplifies RealNVP.

3.3 Auto-Encoding (AE) Model

The auto-encoding model's goal is to reconstruct (part of) inputs from (corrupted) inputs. Due to its flexibility, the AE model is probably the most popular generative model with many variants.

3.3.1 Basic AE Model

Autoencoder (AE) was first introduced in [8] for pre-training artificial neural networks. Before autoencoder, Restricted Boltzmann Machine (RBM) [109] can also be viewed as a special "autoencoder". RBM is an undirected graphical model, and it only contains two layers: the visible layer and the hidden layer. The objective of RBM is to minimize the difference between the marginal distribution of models and data distributions. In contrast, an autoencoder can be regarded as a directed graphical model, and it can be trained more efficiently. Autoencoder is typically for dimensionality reduction. Generally, the autoencoder is a feed-forward neural network trained to produce its input at the output layer. The AE is comprised of an *encoder* network $h = f_{enc}(x)$ and a *decoder* network $x' = f_{dec}(h)$. The objective of AE is to make x and x' as similar as possible (such as through mean-square error). It can be proved that the linear autoencoder corresponds to the PCA method. Sometimes the number of hidden units is greater than the number of input units, and some interesting structures can be discovered by imposing sparsity constraints on the hidden units [85].

3.3.2 Context Prediction Model (CPM)

The idea of the Context Prediction Model (CPM) is to predict contextual information based on inputs.

In NLP, when it comes to self-supervised learning on word embedding, CBOW and Skip-Gram [80] are pioneering works. CBOW aims to predict the input tokens based on context tokens. In contrast, Skip-Gram aims to predict context tokens based on input tokens. Usually, negative sampling is employed to ensure computational efficiency and scalability. Following CBOW architecture, FastText [10] is proposed by utilizing subword information.

Inspired by the progress of word embedding models in NLP, many network embedding models are proposed based on a similar context prediction objective. Deepwalk [92] samples truncated random walks to learn latent node embedding based on the Skip-Gram model. It treats random walks as the equivalent of sentences. However, another network embedding approach LINE [115] aims to

generate neighbors rather than nodes on a path based on current nodes

$$O = - \sum_{(i,j) \in E} w_{ij} \log p(v_j|v_i), \qquad (4)$$

where E denotes edge set, v denotes the node, w_{ij} represents the weight of edge (v_i, v_j). LINE also uses negative sampling to sample multiple negative edges to approximate the objective.

3.3.3 Denoising AE Model

The intuition of denoising autoencoder models is that representation should be robust to the introduction of noise. The masked language model (MLM), one of the most successful architectures in natural language processing, can be regarded as a denoising AE model. To model text sequence, the masked language model (MLM) randomly masks some of the tokens from the input and then predicts them based on their context information, which is similar to the *Cloze* task [117]. BERT [27] is the most representative work in this field. Specifically, in BERT, a unique token [MASK] is introduced in the training process to mask some tokens. However, one shortcoming of this method is that there are no input [MASK] tokens for down-stream tasks. To mitigate this, the authors do not always replace the predicted tokens with [MASK] in training. Instead, they replace them with original words or random words with a small probability.

Following BERT, many extensions of MLM emerge. SpanBERT [58] chooses to mask continuous random spans rather than random tokens adopted by BERT. Moreover, it trains the span boundary representations to predict the masked spans, inspired by ideas in coreference resolution. ERNIE (Baidu) [114] masks entities or phrases to learn entity-level and phrase-level knowledge, which obtains good results in Chinese natural language processing tasks. ERNIE (Tsinghua) [149] further integrates knowledge (entities and relations) in knowledge graphs into language models.

Compared with the AR model, in denoising AE for language modeling, the predicted tokens have access to contextual information from both sides. However, the fact that MLM assumes the predicted tokens are independent if the unmasked tokens are given (which does not hold in reality) has long been considered as its inherent drawback.

In graph learning, Hu *et al.* [52] proposes GPT-GNN, a generative pre-training method for graph neural networks. It also leverages the graph masking techniques and then asks the graph neural network to generate masked edges and attributes. GPT-GNN's wide range of experiments on OAG [108], [116], [144], the largest public, academic graph with 100 million nodes and 2 billion edges, shows impressive improvements on various graph learning tasks.

3.3.4 Variational AE Model

The variational auto-encoding model assumes that data are generated from underlying latent (unobserved) representation. The posterior distribution over a set of unobserved variables $Z = \{z_1, z_2, \ldots, z_n\}$ given some data X is approximated by a variational distribution $q(z|x) \approx p(z|x)$. In variational inference, the evidence lower bound (ELBO) on the

log-likelihood of data is maximized during training.

$$\log p(x) \geq -D_{KL}(q(z|x)||p(z)) + \mathbb{E}_{\sim q(z|x)}[\log p(x|z)], \qquad (5)$$

where $p(x)$ is evidence probability, $p(z)$ is prior and $p(x|z)$ is likelihood probability. The right-hand side of the above equation is called ELBO. From the auto-encoding perspective, the first term of ELBO is a regularizer forcing the posterior to approximate the prior. The second term is the likelihood of reconstructing the original input data based on latent variables.

Variational Autoencoders (VAE) [63] is one important example where variational inference is utilized. VAE assumes the prior $p(z)$ and the approximate posterior $q(z|x)$ both follow Gaussian distributions. Specifically, let $p(z) \sim \mathcal{N}(0, 1)$. Furthermore, reparameterization trick is utilized for modeling approximate posterior $q(z|x)$. Assume $z \sim \mathcal{N}(\mu, \sigma^2)$, $z = \mu + \sigma\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$. Both μ and σ are parameterized by neural networks. Based on calculated latent variable z, decoder network is utilized to reconstruct the input data.

Recently, a novel and powerful variational AE model called VQ-VAE [123] was proposed. VQ-VAE aims to learn discrete latent variables motivated by the fact that many modalities are inherently discrete, such as language, speech, and images. VQ-VAE relies on vector quantization (VQ) to learn the posterior distribution of discrete latent variables. The discrete latent variables are calculated by the nearest neighbor lookup using a shared, learnable embedding table. In training, the gradients are approximated through straight-through estimator [9] as

$$\mathcal{L}(x, D(e)) = \|x - D(e)\|_2^2 + \|sg[E(x)] - e\|_2^2 + \beta\|sg[e] - E(x)\|_2^2, \qquad (6)$$

where e refers to the codebook, the operator sg refers to a stop-gradient operation that blocks gradients from flowing into its argument, and β is a hyperparameter which controls the reluctance to change the code corresponding to the encoder output.

More recently, researchers propose VQ-VAE-2 [101], which can generate versatile high-fidelity images that rival BigGAN [11] on ImageNet [26], the state-of-the-art GAN model. First, the authors enlarge the scale and enhance the autoregressive priors by a powerful PixelCNN [122] prior. Additionally, they adopt a multi-scale hierarchical organization of VQ-VAE, which enables learning local information and global information of images separately. Nowadays, VAE and its variants have been widely used in the computer vision area, such as image representation learning, image generation, video generation.

Variational auto-encoding models have also been employed in node representation learning on graphs. For example, Variational graph auto-encoder (VGAE) [65] uses the same variational inference technique as VAE with graph convolutional networks (GCN) [64] as the encoder. Due to the uniqueness of graph-structured data, the objective of VGAE is to reconstruct the adjacency matrix of the graph by measuring node proximity. Zhu *et al.* [150] propose DVNE, a deep variational network embedding model in Wasserstein space. It learns Gaussian node embedding to model the uncertainty of nodes. 2-Wasserstein distance is used to

measure the similarity between the distributions for its effectiveness in preserving network transitivity. vGraph [111] can perform node representation learning and community detection collaboratively through a generative variational inference framework. It assumes that each node can be generated from a mixture of communities, and each community is defined as a multinomial distribution over nodes.

3.4 Hybrid Generative Models

3.4.1 Combining AR and AE Model

Some researchers propose to combine the advantages of both AR and AE. MADE [78] makes a simple modification to autoencoder. It masks the autoencoder's parameters to respect auto-regressive constraints. Specifically, for the original autoencoder, neurons between two adjacent layers are fully-connected through MLPs. However, in MADE, some connections between adjacent layers are masked to ensure that each input dimension is reconstructed solely from its dimensions. MADE can be easily parallelized on conditional computations, and it can get direct and cheap estimates of high-dimensional joint probabilities by combining AE and AR models.

In NLP, Permutation Language Model (PLM) [138] is a representative model that combines the advantage of auto-regressive model and auto-encoding model. XLNet [138], which introduces PLM, is a generalized auto-regressive pre-training method. XLNet enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. To formalize the idea, let $\mathcal{Z}_T$ denotes the set of all possible permutations of the length-T index sequence $[1, 2, \ldots, T]$, the objective of PLM can be expressed as follows:

$$\max_{\theta} \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_T} \left[\sum_{t=1}^{T} \log p_\theta(x_{z_t} | \mathbf{x}_{\mathbf{z}_{<t}}) \right]. \tag{7}$$

Actually, for each text sequence, different factorization orders are sampled. Therefore, each token can see its contextual information from both sides. Based on the permuted order, XLNet also conducts reparameterization with positions to let the model know which position is needed to predict. Then a special two-stream self-attention is introduced for target-aware prediction.

Furthermore, different from BERT, inspired by the latest advancements in the AR model, XLNet integrates the segment recurrence mechanism and relative encoding scheme of Transformer-XL [24] into pre-training, which can model long-range dependency better than Transformer [125].

3.4.2 Combining AE and Flow-Based Models

In the graph domain, GraphAF [107] is a flow-based auto-regressive model for molecule graph generation. It can generate molecules in an iterative process and also calculate the exact likelihood in parallel. GraphAF formalizes molecule generation as a sequential decision process. It incorporates detailed domain knowledge into the reward design, such as valency check. Inspired by the recent progress of flow-based models, it defines an invertible transformation from a base distribution (e.g., multivariate Gaussian) to a molecular graph structure. Additionally, Dequantization technique [50]

is utilized to convert discrete data (including node types and edge types) into continuous data.

3.5 Pros and Cons

A reason for the generative self-supervised learning's success in self-supervised learning is its ability to recover the original data distribution without assumptions for downstream tasks, which enables generative models' wide applications in both classification and generation. Notably, all the existing generation tasks (including text, image, and audio) rely heavily on generative self-supervised learning. Nevertheless, two shortcomings restrict its performance.

First, despite its central status in generation tasks, generative self-supervised learning is recently found far less competitive than contrastive self-supervised learning in some classification scenarios because contrastive learning's goal naturally conforms the classification objective. Works including MoCo [47], SimCLR [15], BYOL [42] and SwAV [14] have presented overwhelming performances on various CV benchmarks. Nevertheless, in the NLP domain, researchers still depend on generative language models to conduct text classification.

Second, the point-wise nature of the generative objective has some inherent defects. This objective is usually formulated as a maximum likelihood function $\mathcal{L}_{MLE} = -\sum_x \log p(x|c)$ where x is all the samples we hope to model, and c is a conditional constraint such as context information. Considering its form, MLE has two fatal problems:

1) *Sensitive and Conservative Distribution*. When $p(x|c) \to 0$, $\mathcal{L}_{MLE}$ becomes super large, making generative model extremely sensitive to rare samples. It directly leads to a conservative distribution, which has a low performance.

2) *Low-level Abstraction Objective*. In MLE, the representation distribution is modeled at x's level (i.e., point-wise level), such as pixels in images, words in texts, and nodes in graphs. However, most of the classification tasks target at *high-level abstraction*, such as object detection, long paragraph understanding, and molecule classification.

and as an opposite approach, generative-contrastive self-supervised learning abandons the point-wise objective. It turns to distributional matching objectives that are more robust and better handle the high-level abstraction challenge in the data manifold.

4 CONTRASTIVE SELF-SUPERVISED LEARNING

From a statistical perspective, machine learning models are categorized into generative and discriminative models. Given the joint distribution $P(X, Y)$ of the input X and target Y, the generative model calculates the $p(X|Y = y)$ while the discriminative model tries to model the $P(Y|X = x)$. Because most of the representation learning tasks hope to model relationships between x, for a long time, people believe that the generative model is the only choice for representation learning.

However, recent breakthroughs in contrastive learning, such as Deep InfoMax, MoCo and SimCLR, shed light on the potential of discriminative models for representation. Contrastive learning aims at "learn to compare" through a
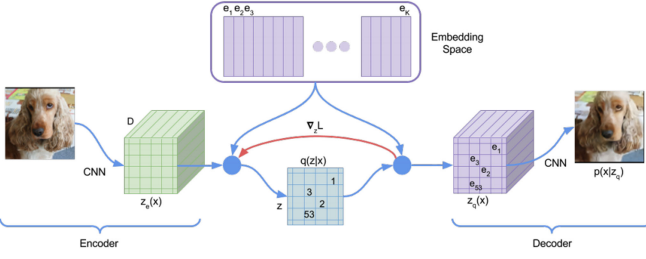
Fig. 5. Architecture of VQ-VAE [123]. Compared to VAE, the orginal hidden distribution is replaced with a quantized vector dictionary. In addition, the prior distribution is replaced with a pre-trained PixelCNN that models the hierarchical features of images. Taken from [123].

Noise Contrastive Estimation (NCE) [44] objective formatted as

$$\mathcal{L} = \mathbb{E}_{x,x^+,x^-}\left[-log\left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + e^{f(x)^T f(x^-)}}\right)\right], \quad (8)$$

where x^+ is similar to x, x^- is dissimilar to x and f is an encoder (representation function). The similarity measure and encoder may vary from task to task, but the framework remains the same. With more dissimilar pairs involved, we have the InfoNCE [88] formulated as

$$\mathcal{L} = \mathbb{E}_{x,x^+,x^k}\left[-log\left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_{k=1}^K e^{f(x)^T f(x^k)}}\right)\right]. \quad (9)$$

Here we divide recent contrastive learning frameworks into 2 types: *context-instance* contrast and *instance-instance* contrast. Both of them achieve amazing performance in downstream tasks, especially on classification problems under the linear protocol.

4.1 *Context-Instance* Contrast

The context-instance contrast, or so-called *global-local* contrast, focuses on modeling the belonging relationship between the local feature of a sample and its global context representation. When we learn the representation for a local feature, we hope it is associative to the representation of the global content, such as stripes to tigers, sentences to its paragraph, and nodes to their neighborhoods.

There are two main types of Context-Instance Contrast: Predict Relative Position (PRP) and Maximize Mutual Information (MI). The differences between them are:

- PRP focuses on learning relative positions between local components. The global context serves as an implicit requirement for predicting these relations
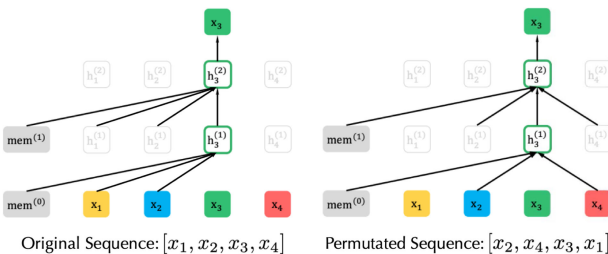

Fig. 6. Illustration for permutation language modeling [138] objective for predicting x_3 given the same input sequence x but with different factorization orders. Adapted from [138].
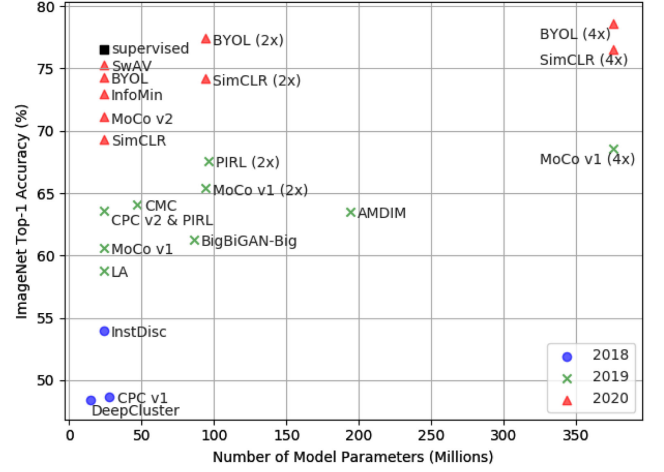

Fig. 7. Self-supervised representation learning performance on ImageNet top-1 accuracy in March, 2021, under linear classification protocol. The self-supervised learning's ability on feature extraction is rapidly approaching the supervised method (ResNet50). Except for BigBiGAN, all the models above are contrastive self-supervised learning methods.

(such as understanding what an elephant looks like is critical for predicting relative position between its head and tail).

- MI focuses on learning the direct belonging relationships between local parts and global context. The relative positions between local parts are ignored.

4.1.1 Predict Relative Position

Many data contain rich spatial or sequential relations between parts of it. For example, in image data such as Fig. 8, the elephant's head is on the *right* of its tail. In text data, a sentence like "Nice to meet you." would probably be ahead of "Nice to meet you, too.". Various models regard recognizing relative positions between parts of it as the pretext task [57]. It could be to predict relative positions of two patches from a sample [32], or to recover positions of shuffled segments of an image (solve jigsaw) [61], [86], [131], or to infer the rotation angle's degree of an image [38]. PRP may also serve as tools to create hard positive samples. For instance, the jigsaw technique is applied in PIRL [81] to augment the positive sample, but PIRL does not regard solving jigsaw and recovering spatial relation as its objective.

In the pre-trained language model, similar ideas such as Next Sentence Prediction (NSP) are also adopted. NSP loss is initially introduced by BERT [27], where for a sentence, the model is asked to distinguish the following and a randomly sampled one. However, some later work empirically proves that NSP helps little, even harm the performance. So in RoBERTa [75], the NSP loss is removed.
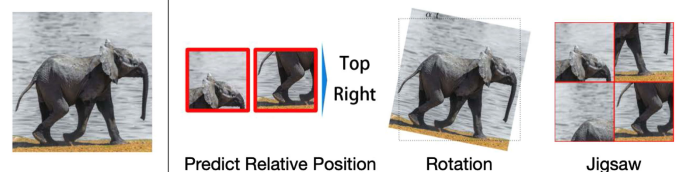

Fig. 8. Three typical methods for spatial relation contrast: predict relative position [32], rotation [38] and solve jigsaw [61], [81], [86], [131].

To replace NSP, ALBERT [68] proposes Sentence Order Prediction (SOP) task. That is because, in NSP, the negative next sentence is sampled from other passages that may have different topics from the current one, turning the NSP into a far easier topic model problem. In SOP, two sentences that exchange their position are regarded as a negative sample, making the model concentrate on the semantic meaning's coherence.

4.1.2 Maximize Mutual Information

This kind of method derives from mutual information (MI) – a fundamental concept in statistics. Mutual information targets modeling the association between two variables, and our objective is to maximize it. Generally, this kind of models optimize

$$\max_{g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_1} I(g_1(x_1), g_2(x_2)), \quad (10)$$

where g_i is the representation encoder, $\mathcal{G}_i$ is a class of encoders with some constraints, and $I(\cdot, \cdot)$ is a sample-based estimator for the accurate mutual information. In applications, MI is notorious for its complex computation. A common practice is to alternatively maximize I's lower bound with an NCE objective.

Deep InfoMax [49] is the first one to explicitly model mutual information through a contrastive learning task, which maximize the MI between a local patch and its global context. For real practices, take image classification as an example, we can encode a cat image x into $f(x) \in \mathbb{R}^{M \times M \times d}$, and take out a local feature vector $v \in \mathbb{R}^d$. To conduct contrast between instance and context, we need two other things:

- a summary function $g : \mathbb{R}^{M \times M \times d} \to \mathbb{R}^d$ to generate the context vector $s = g(f(x)) \in \mathbb{R}^d$
- another cat image x^- and its context vector $s^- = g(f(x^-))$.

and the contrastive objective is then formulated as

$$\mathcal{L} = \mathbb{E}_{v,x} \left[-\log \left(\frac{e^{v^T \cdot s}}{e^{v^T \cdot s} + e^{v^T \cdot s^-}} \right) \right]. \quad (11)$$

Deep InfoMax provides us with a new paradigm and boosts the development of self-supervised learning. The first influential follower is Contrastive Predictive Coding (CPC) [88] for speech recognition. CPC maximizes the association between a segment of audio and its context audio. To improve data efficiency, it takes several negative context vectors at the same time. Later on, CPC has also been applied in image classification.

AMDIM [6] enhances the positive association between a local feature and its context. It randomly samples two different views of an image (truncated, discolored, and so forth) to generate the local feature vector and context vector, respectively. CMC [118] extends it into several different views for one image and samples another irrelevant image as the negative. However, CMC is fundamentally different from Deep InfoMax and AMDIM because it proposes to measure the instance-instance similarity rather than context-instance similarity. We will discuss it in the following subsection.

In language pre-training, InfoWord [66] proposes to maximize the mutual information between a global representation of a sentence and *n-grams* in it. The context is induced from the sentence with selected n-grams being masked, and the negative contexts are randomly picked out from the corpus.

In graph learning, Deep Graph InfoMax (DGI) [127] regards a node's representation as the local feature and the average of randomly samples 2-hop neighbors as the context. However, it is hard to generate negative contexts on a single graph. To solve this problem, DGI proposes to *corrupt* the original context by keeping the sub-graph structure and permuting the node features. DGI is followed by many works, such as InfoGraph [110], which targets learning graph-level representation rather than node level, maximizing the mutual information between graph-level representation and substructures at different levels. As what CMC has done to improve Deep InfoMax, in [46] authors propose a contrastive multi-view representation learning method for the graph. They also discover that graph diffusion is the most effective way to yield augmented positive sample pairs in graph learning.

As an attempt to unify graph pre-training, in [51], the authors systematically analysis the pre-training strategies for graph neural networks from two dimensions: attribute/structural and node-level/graph-level. For structural prediction at node-level, they propose Context Prediction to maximize the MI between the k-hop neighborhood's representations and its context graph. For attributes in the chemical domain, they propose Attribute Mask to predict a node's attribute according to its neighborhood, which is a generative objective similar to token masks in BERT.

S^{2}GRL [91] further separates nodes in the context graph into k-hop context subgraphs and maximizes their MI with target node, respectively. However, a fundamental problem of graph pre-training is about learning inductive biases across graphs, and existing graph pre-training work is only applicable for a specific domain.

4.2 Instance-Instance Contrast

Though MI-based contrastive learning achieves great success, some recent studies [15], [18], [47], [120] cast doubt on the actual improvement brought by MI.

The [120] provides empirical evidence that the success of the models mentioned above is only loosely connected to MI by showing that an upper bound MI estimator leads to ill-conditioned and lower performance representations. Instead, more should be attributed to encoder architecture and a negative sampling strategy related to metric learning. A significant focus in metric learning is to perform hard positive sampling while increasing the negative sampling efficiency. They probably play a more critical role in MI-based models' success.

As an alternative, instance-instance contrastive learning discards MI and directly studies the relationships between different samples' instance-level local representations as what metric learning does. Instance-level representation, rather than context-level, is more crucial for a wide range of classification tasks. For example, in an image classified as "dog", while there must be dog instances, some other

irrelevant context objects such as grass might appear. But what matters for the image classification is the dog rather than the context. Another example would be sentence emotional classification, which primarily relies on few but important keywords.

In the early stage of instance-instance contrastive learning's development, researchers borrow ideas from semi-supervised learning to produce pseudo labels via cluster-based discrimination and achieve rather good performance on representations. More recently, CMC [118], MoCo [47], SimCLR [15], and BYOL [42] further support the above conclusion by outperforming the context-instance contrastive methods and achieve a competitive result to supervised methods under the linear classification protocol. We will start with cluster-based discrimination proposed earlier and then turn to instance-based discrimination.

4.2.1 Cluster Discrimination

Instance-instance contrast is first studied in clustering-based methods [13], [73], [87], [137], especially the DeepCluster [13] which first achieves competitive performance to the supervised model AlexNet [67].

Image classification asks the model to categorize images correctly, and the representation of images in the same category should be similar. Therefore, the motivation is to pull similar images near in the embedding space. In supervised learning, this pulling-near process is accomplished via label supervision; in self-supervised learning, however, we do not have such labels. To solve the label problem, Deep Cluster [13] proposes to leverage clustering to yield pseudo labels and asks a discriminator to predict images' labels. The training could be formulated in two steps. In the first step, DeepCluster uses K-means to cluster encoded representation and produces pseudo labels for each sample. Then in the second step, the discriminator predicts whether two samples are from the same cluster and back-propagates to the encoder. These two steps are performed iteratively.

Recently, Local Aggregation (LA) [152] has pushed forward the cluster-based method's boundary. It points out several drawbacks of DeepCluster and makes the corresponding optimization. First, in DeepCluster, samples are assigned to mutual-exclusive clusters, but LA identifies neighbors separately for each example. Second, DeepCluster optimizes a cross-entropy discriminative loss, while LA employs an objective function that directly optimizes a local soft-clustering metric. These two changes substantially boost the performance of LA representation on downstream tasks.

A similar work to LA would be VQ-VAE [101], [123] that we introduce in Section 3. To conquer the traditional deficiency for VAE to generate high-fidelity images, VQ-VAE proposes quantizing vectors. For the feature matrix encoded from an image, VQ-VAE substitutes each 1-dimensional vector in the matrix to the nearest one in an embedding dictionary. This process is somehow the same as what LA is doing.

Clustering-based discrimination may also help in the generalization of other pre-trained models, transferring models from pretext objectives to downstream tasks better. Traditional representation learning models have only two

stages: one for pre-training and the other for evaluation. ClusterFit [136] introduces a cluster prediction fine-tuning stage similar to DeepCluster between the above two stages, which improves the representation's performance on downstream classification evaluation.

Despite the previous success of cluster discrimination-based contrastive learning, the two-stage training paradigm is time-consuming and poor performing compared to later instance discrimination-based methods, including CMC [118], MoCo [47] and SimCLR [15]. These instance discrimination-based methods have got rid of the slow clustering stage and introduced efficient data augmentation (i.e., multi-view) strategies to boost the performance. In light of these problems, authors in SwAV [14] bring online clustering ideas and multi-view data augmentation strategies into the cluster discrimination approach. SwAV proposes a swapped prediction contrastive objectives to deal with multi-view augmentation. The intuition is that, given some (clustered) prototypes, different views of the same images should be assigned into the same prototypes. SwAV names this "assignment" as "codes". To accelerate code computing, the authors of SwAV design an online computing strategy. SwAV outperforms instance discrimination-based methods when model size is small and is more computationally efficient. Based on SwAV, a 1.3-billion-parameter SEER [41] is trained on 1 billion web images collected from Instagram.

In graph learning, M3S [113] adopts a similar idea to perform DeepCluster-style self-supervised pre-training for better semi-supervised prediction. Given little labeled data and many unlabeled data, for every stage, M3S first pre-train itself to produce pseudo labels on unlabeled data as DeepCluster does and then compares these pseudo labels with those predicted by the model being supervised trained on labeled data. Only top-k confident labels are added into a labeled set for the next stage of semi-supervised training. In [143], this idea is further developed into three pre-training tasks: topology partitioning (similar to spectral clustering), node feature clustering, and graph completion.

4.2.2 Instance Discrimination

The prototype of leveraging instance discrimination as a pretext task is InstDisc [132]. Based on InstDisc, CMC [118] proposes to adopt multiple different views of an image as positive samples and take another one as the negative. CMC draws near multiple views of an image in the embedding space and pulls away from other samples. However, it is somehow constrained by the idea of Deep InfoMax, which only samples one negative sample for each positive one.

In MoCo [47], researchers further develop the idea of leveraging *instance discrimination* via momentum contrast, which substantially increases the amount of negative samples. For example, given an input image x, our intuition is to learn a instinct representation $q = f_q(x)$ by a query encoder $f_q(\cdot)$ that can distinguish x from any other images. Therefore, for a set of other images x_i, we employ an asynchronously updated key encoder $f_k(\cdot)$ to yield $k_+ = f_k(x)$ and $k_i = f_k(x_i)$, and optimize the following objective
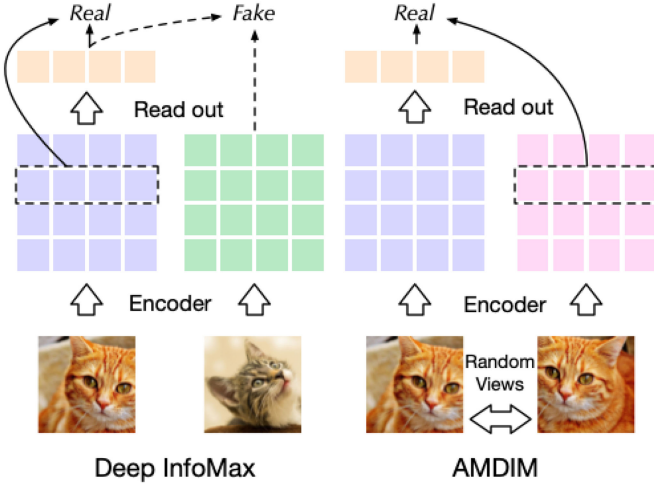
Fig. 9. Two representatives for mutual information's application in contrastive learning. Deep InfoMax (DIM) [49] first encodes an image into feature maps, and leverage a read-out function (or so-called summary function) to produce a summary vector. AMDIM [6] enhances the DIM through randomly choosing another view of the image to produce the summary vector.

$$\mathcal{L} = -\log \frac{exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} exp(q \cdot k_i/\tau)}, \qquad (12)$$

where K is the number of negative samples. This formula is in the form of InfoNCE.

Besides, MoCo presents two other critical ideas in dealing with negative sampling efficiency.

- First, it abandons the traditional end-to-end training framework. It designs the momentum contrast learning with two encoders (query and key), which prevents the fluctuation of loss convergence in the beginning period.
- Second, to enlarge negative samples' capacity, MoCo employs a queue (with K as large as 65536) to save the recently encoded batches as negative samples. This significantly improves the negative sampling efficiency.

There are some other auxiliary techniques to ensure the training convergence, such as batch shuffling to avoid trivial solutions and temperature hyper-parameter τ to adjust the scale.

However, MoCo adopts a too simple positive sample strategy: a pair of positive representations come from the same sample without any transformation or augmentation,
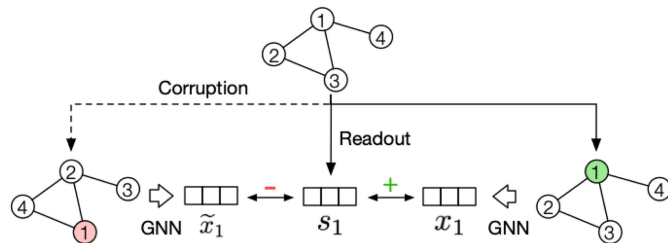

Fig. 10. Deep Graph InfoMax [127] uses a readout function to generate summary vector s_1, and puts it into a discriminator with node 1's embedding x_1 and corrupted embedding $\widetilde{x}_1$ respectively to identify which embedding is the real embedding. The corruption is to shuffle the positions of nodes.
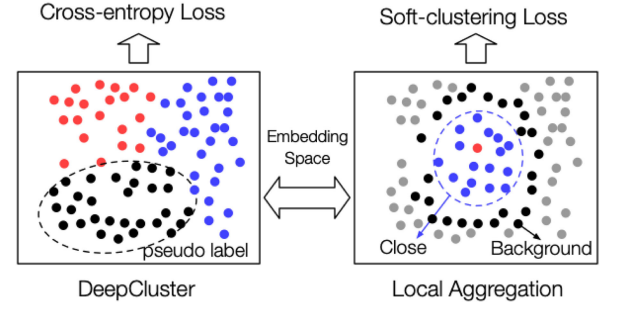

Fig. 11. Cluster-based instance-instance contrastive emthods: Deep-Cluster [13] and Local Aggregation [152]. In the embedding space, DeepCluster uses clustering to yield pseudo labels for discrimination to draw near similar samples. However, Local Aggregation shows that a egocentric soft-clustering objective would be more effective.

making the positive pair far too easy to distinguish. PIRL [81] adds jigsaw augmentation as described in Section 4.1.1. PIRL asks the encoder to regard an image and its jigsawed one as similar pairs to produce a pretext-invariant representation.

In SimCLR [15], the authors further illustrate the importance of a hard positive sample strategy by introducing data augmentation in 10 forms. This data augmentation is similar to CMC [118], which leverages several different views to augment the positive pairs. SimCLR follows the end-to-end training framework instead of momentum contrast from MoCo, and to handle the large-scale negative samples problem, SimCLR chooses a batch size of N as large as 8196.

The details are as follows. A minibatch of N samples is augmented to be $2N$ samples $\hat{x}_j(j = 1, 2, \ldots, 2N)$. For a pair of a positive sample $\hat{x}_i$ and $\hat{x}_j$ (derive from one original sample), other $2(N - 1)$ are treated as negative ones. A pairwise contrastive loss NT-Xent loss [17] is defined as

$$l_{i,j} = -\log \frac{exp(sim(\hat{x}_i, \hat{x}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} exp(sim(\hat{x}_i, \hat{x}_k)/\tau)}, \qquad (13)$$

noted that $l_{i,j}$ is asymmetrical, and the $sim(\cdot, \cdot)$ function here is a cosine similarity function that can normalize the representations. The summed up loss is

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [l_{2i-1,2i} + l_{2i,2i-1}]. \qquad (14)$$

SimCLR also provides some other practical techniques, including a learnable nonlinear transformation between the representation and the contrastive loss, more training steps, and deeper neural networks. [18] conducts ablation studies to show that techniques in SimCLR can also further improve MoCo's performance.

More investigation into augmenting positive samples is made in InfoMin [119]. The authors claim that we should select those views with less mutual information for better-augmented views in contrastive learning. In the optimal situation, the views should only share the label information. To produce such optimal views, the authors first propose an unsupervised method to minimize mutual information between views. However, this may result in a loss of information for predicting labels (such as a pure blank view). Therefore, a semi-supervised method is then proposed to
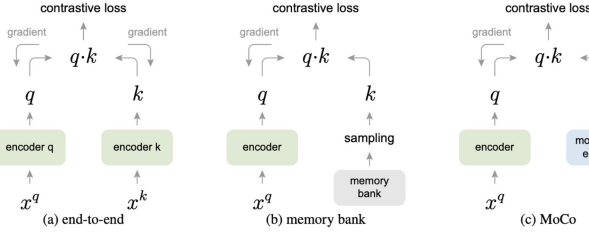
Fig. 12. Conceptual comparison of three contrastive loss mechanisms. Taken from MoCo [47].
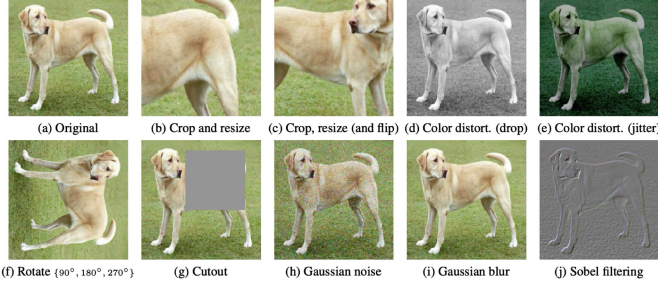
Fig. 13. Ten different views adopted by SIMCLR [15]. The enhancement of positive samples substantially improves the self-supervised learning performance. Taken from [15].
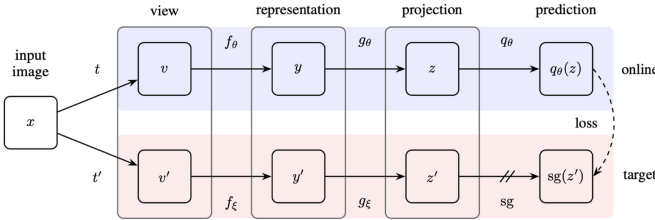
Fig. 14. The architecture of BYOL [42]. Noted that the online encoder has an additional layer q_θ compared to the target one, which gives the representations some flexibility to be improved during the training. Taken from [42].

find views sharing only label information. This technique leads to an improve about 2% over MoCo v2.

A more radical step is made by BYOL [42], which discards negative sampling in self-supervised learning but achieves an even better result over InfoMin. For contrastive learning methods we mentioned above, they learn representations by predicting different views of the same image and cast the prediction problem directly in representation space. However, predicting directly in representation space can lead to collapsed representations because multi-views are generally *too predictive* for each other. Without negative samples, it would be too easy for the neural networks to distinguish those positive views.

In BYOL, researchers argue that negative samples may not be necessary in this process. They show that, if we use a fixed randomly initialized network (which would not collapse because it is not trained) to serve as the key encoder, the representation produced by query encoder would still be improved during training. If then we set the target encoder to be the trained query encoder and iterate this procedure, we would progressively achieve better performance. Therefore, BYOL proposes an architecture (Fig. 14) with an exponential moving average strategy to update the target encoder just as MoCo does. Additionally, instead of
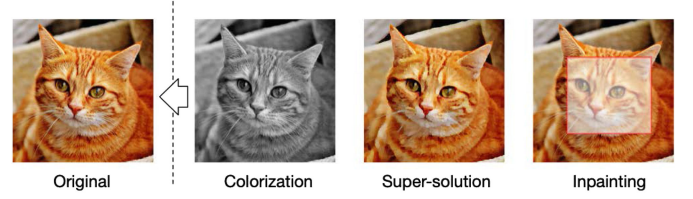

Fig. 15. Illustration of typical "recovering with partial input" methods: colorization, inpainting and super-resolution. Given the original input on the left, models are asked to recover it with different partial inputs given on the right.
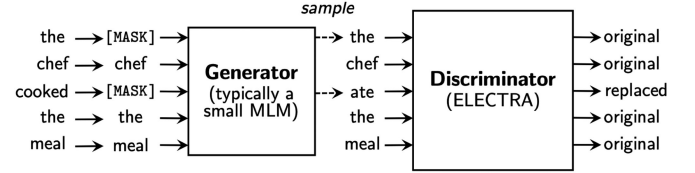

Fig. 16. The architecture of ELECTRA [21]. It follows GAN's framework but uses a two-stage training paradigm to avoid using policy gradient. The MLM is Masked Language Model.
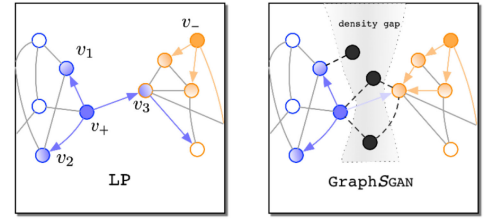

Fig. 17. The architecture of GraphSGAN [28], which investigates density gaps in embedding space for classification problems. Taken from [28].

using cross-entropy loss, they follow the regression paradigm in which mean square error is used as

$$\mathcal{L}_\theta^{\text{BYOL}} \triangleq ||\bar{q}_\theta(z_\theta) - \bar{z}_\xi'||_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z_\xi' \rangle}{||q_\theta(z_\theta)||_2 \cdot ||z_\xi'||_2}. \quad (15)$$

This not only makes the model better-performed in downstream tasks, but also more robust to smaller batch size. In MoCo and SimCLR, a drop in batch size results in a significant decline in performance. However, in BYOL, although batch size still matters, it is far less critical. The ablation study shows that a batch size of 512 only causes a drop of 0.3% compared to a standard batch size of 4,096, while SimCLR shows a drop of 1.4%.

In SimSiam [19], researchers further study how necessary is negative sampling, and even batch normalization in contrastive representation learning. They show that the most critical component in BYOL is the *stop gradient* operation, which makes the target representation stable. SimSiam is proved to converge faster than MoCo, SimCLR, and BYOL with even smaller batch sizes, while the performance only slightly decreases.

Some other works are inspired by theoretical analysis into the contrastive objective. ReLIC [82] argues that contrastive pre-training teaches the encoder to causally disentangle the invariant content (i.e., main objects) and style (i.e., environments) in an image. To better enforce this observation in the data augmentation, they propose to add an extra KL-divergence regularizer between prediction

logits of an image's different views. The results show that this can enhance the models' generalization ability and robustness and improve the performance.

In graph learning, Graph Contrastive Coding (GCC) [94] is a pioneer to leverage instance discrimination as the pretext task for structural information pre-training. For each node, we sample two subgraphs independently by random walks with restart and use top eigenvectors from their normalized graph Laplacian matrices as nodes' initial representations. Then we use GNN to encode them and calculate the InfoNCE loss as what MoCo and SimCLR do, where the node embeddings from the same node (in different subgraphs) are viewed as similar. Results show that GCC learns better transferable structural knowledge than previous work such as struc2vec [103], GraphWave [35] and ProNE [145]. GraphCL [142] studies the data augmentation strategies in graph learning. They propose four different augmentation methods based on edge perturbation and node dropping. It further demonstrates that the appropriate combination of these strategies can yield even better performance.

4.3 Self-Supervised Contrastive Pre-Training for Semi-Supervised Self-Training

While contrastive learning-based self-supervised learning continues to push the boundaries on various benchmarks, labels are still important because there is a gap between training objectives of self-supervised learning and supervised learning. In other words, no matter how self-supervised learning models improve, they are still the only powerful feature extractor, and to transfer to the downstream task, we still need labels more or less. As a result, to bridge the gap between self-supervised pre-training and downstream tasks, semi-supervised learning is what we are looking for.

Recall the MoCo [47] that have topped the ImageNet leader-board. Although it is proved beneficial for many other downstream vision tasks, it fails to improve the COCO object detection task. Some following work [84], [153] investigates this problem and attributes it to the gap between the instance discrimination and object detection. In such a situation, while pure self-supervised pre-training fails to help, semi-supervised-based self-training can contribute a lot to it.

First, we will clarify the definitions of semi-supervised learning and self-training. Semi-supervised learning is an approach to machine learning that combines a small amount of labeled data with many unlabeled data during training. Various methods derive from several different assumptions made on the data distribution, with self-training (or self-labeling) being the oldest. In self-training, a model is trained on the small amount of labeled data and then yield labels on unlabeled data. Only those data with highly confident labels are combined with original labeled data to train a new model. We iterate this procedure to find the best model.

The current state-of-the-art supervised model [133] on ImageNet follows the self-training paradigm, where we first train an EfficientNet model on labeled ImageNet images and use it as a teacher to generate pseudo labels on 300M unlabeled images. We then train a larger EfficientNet as a student model based on labeled and pseudo labeled images. We iterate this process by putting back the student as the teacher. During the pseudo labels generation, the teacher is not noised so that the pseudo labels are as accurate as possible. However, during the student's learning, we inject noise such as dropout, stochastic depth, and data augmentation via RandAugment to the student to generalize better than the teacher.

In light of semi-supervised self-training's success, it is natural to rethink its relationship with the self-supervised methods, especially with the successful contrastive pre-trained methods. In Section 4.2.1, we have introduced M3S [112] that attempts to combine cluster-based contrastive pre-training and downstream semi-supervised learning. For computer vision tasks, Zoph et al. [153] study the MoCo pre-training and a self-training method in which a teacher is first trained on a downstream dataset (e.g., COCO) and then yield pseudo labels on unlabeled data (e.g., ImageNet), and finally a student learns jointly over real labels on the downstream dataset and pseudo labels on unlabeled data. They surprisingly find that pre-training's performance hurts while self-training still benefits from strong data augmentation. Besides, more labeled data diminishes the value of pre-training, while semi-supervised self-training always improves. They also discover that the improvements from pre-training and self-training are orthogonal to each other, i.e., contributing to the performance from different perspectives. The model with joint pre-training and self-training is the best.

Chen et al. [16]'s SimCLR v2 supports the conclusion mentioned above by showing that with only 10% of the original ImageNet labels, the ResNet-50 can surpass the supervised one with joint pre-training and self-training. They propose a 3-step framework:

1) Do self-supervised pre-training as SimCLR v1, with some minor architecture modification and a deeper ResNet.
2) Fine-tune the last few layers with only 1% or 10% of original ImageNet labels.
3) Use the fine-tuned network as *teacher* to yield labels on unlabeled data to train a smaller *student* ResNet-50.

The success in combining self-supervised contrastive pre-training and semi-supervised self-training opens up our eyes for a future data-efficient deep learning paradigm. More work is expected for investigating their latent mechanisms.

4.4 Pros and Cons

Because contrastive learning has assumed the downstream applications to be classifications, it only employs the encoder and discards the decoder in the architecture compared to generative models. Therefore, contrastive models are usually light-weighted and perform better in discriminative downstream applications.

Contrastive learning is closely related to metric learning, a discipline that has been long studied. However, self-supervised contrastive learning is still an emerging field, and many problems remain to be solved, including:

1) *Scale to natural language pre-training.* Despite its success in computer vision, contrastive pre-training does not present a convincing result in the NLP benchmarks. Most contrastive learning in NLP now lies in BERT's supervised fine-tuning, such as improving BERT's sentence-level representation [102], information retrieval [59]. Few algorithms have been proposed to apply contrastive learning in the pre-training stage. As most language understanding tasks are classifications, a contrastive language pre-training approach should be better than the current generative language models.

2) *Sampling efficiency.* Negative sampling is a must for most contrastive learning, but this process is often tricky, biased, and time-consuming. BYOL [42] and SimSiam [19] are the pioneers to get contrastive learning rid of negative samples, but it can be improved. It is also not clear enough that what role negative sampling plays in contrastive learning.

3) *Data augmentation.* Researchers have proved that data augmentation can boost contrastive learning's performance, but the theory for why and how it helps is still quite ambiguous. This hinders its application into other domains, such as NLP and graph learning, where the data is discrete and abstract.

5 GENERATIVE-CONTRASTIVE (ADVERSARIAL) SELF-SUPERVISED LEARNING

Generative-contrastive representation learning, or in a more familiar name *adversarial representation learning*, leverage discriminative loss function as the objective. Yann Lecun comments on adversarial learning as "the most interesting idea in the last ten years in machine learning.". Its application in learning representation is also booming.

The idea of adversarial learning derives from generative learning, where researchers have observed some inherent shortcomings of point-wise generative reconstruction (See Section 3.5). As an alternative, adversarial learning learns to reconstruct the original data distribution rather than the samples by minimizing the distributional divergence.

In terms of contrastive learning, adversarial methods still preserve the generator structure consisting of an encoder and a decoder. In contrast, the contrastive abandons the decoder component (as shown in Fig. 4). It is critical because, on the one hand, the generator endows adversarial learning with strong expressiveness that is peculiar to generative models; on the other hand, it also makes the objective of adversarial methods far more challenging to learn than that of contrastive methods, leading to unstable convergence. In the adversarial setting, the decoder's existence asks the representation to be "reconstructive," in other words, it contains all the necessary information for constructing the inputs. However, in the contrastive setting, we only need to learn "distinguishable" information to discriminate different samples.

To sum up, the adversarial methods absorb merits from both generative and contrastive methods together with some drawbacks. In a situation where we need to fit on an implicit distribution, it is a better choice. In the following

several subsections, we will discuss its various applications on representation learning.

5.1 Generate With Complete Input

This section introduces GAN and its variants for representation learning, focusing on capturing the sample's complete information.

The inception of adversarial representation learning should be attributed to Generative Adversarial Networks (GAN) [97], which proposes the adversarial training framework. Follow GAN, many variants [11], [55], [56], [60], [72], [89] emerge and reshape people's understanding of deep learning's potential. GAN's training process could be viewed as two players play a game; one generates fake samples while another tries to distinguish them from real ones. To formulate this problem, we define G as the generator, D as the discriminator, $p_{data}(x)$ as the real sample distribution, $p_z(z)$ as the learned latent sample distribution, we want to optimize this min-max game

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))].$$
(16)

Before VQ-VAE2, GAN maintains dominating performance on image generation tasks over purely generative models, such as autoregressive PixelCNN and autoencoder VAE. It is natural to think about how this framework could benefit representation learning.

However, there is a gap between generation and representation. Compared to autoencoder's explicit latent sample distribution $p_z(z)$, GAN's latent distribution $p_z(z)$ is implicitly modeled. We need to extract this implicit distribution out. To bridge this gap, AAE [77] first proposes a solution to follow the autoencoder's natural idea. The generator in GAN could be viewed as an implicit autoencoder. We can replace the generator with an explicit variational autoencoder (VAE) to extract the representation out. Recall the objective of VAE

$$\mathcal{L}_{VAE} = -\mathbb{E}_{q(z|x)}(-\log(p(x|z)) + KL(q(z|x)\|p(z)).$$
(17)

As we mentioned before, compared to $l2$ loss of autoencoder, discriminative loss in GAN better models the high-level abstraction. To alleviate the problem, AAE substitutes the KL divergence function for a discriminative loss

$$\mathcal{L}_{Disc} = \text{CrossEntropy}(q(z), p(z)),$$
(18)

that asks the discriminator to distinguish representation from the encoder and a prior distribution.

However, AAE still preserves the reconstruction error, which contradicts GAN's core idea. Based on AAE, BiGAN [33] and ALI [36] argue to embrace adversarial learning without reservation and put forward a new framework. Given an actual sample x

- Generator G: the generator here virtually acts as the decoder, generates fake samples $x' = G(z)$ by z from a prior latent distribution (e.g., [uniform(-1,1)]^d, d refers to dimension).

- Encoder E: a newly added component, mapping real sample x to representation $z' = E(x)$. This is also exactly what we want to train.

- Discriminator D: given two inputs $[z, G(z)]$ and $[E(x), x]$, decide which one is from the real sample distribution.

It is easy to see that their training goal is $E = G^{-1}$. In other words, encoder E should learn to "convert" generator G. This goal could be rewritten as a l_0 loss for autoencoder [33], but it is not the same as a traditional autoencoder because the distribution does not make any assumption about the data itself. The distribution is shaped by the discriminator, which captures the semantic-level difference. Based on BiGAN and ALI, later studies [20], [34] discover that GAN with deeper and larger networks and modified architectures can produce even better results on downstream task.

5.2 Recover With Partial Input

As we mentioned above, GAN's architecture is not born for representation learning, and modification is needed to apply its framework. While BiGAN and ALI choose to extract the implicit distribution directly, some other methods such as colorization [69], [70], [147], [148], inpainting [55], [89] and super-resolution [72] apply the adversarial learning via in a different way. Instead of asking models to reconstruct the whole input, they provide models with partial input and ask them to recover the rest parts. This is similar to denoising autoencoder (DAE) such as BERT's family in natural language processing but conducted in an adversarial manner.

Colorization is first proposed by [147]. The problem can be described as given one color channel L in an image and predicting the value of two other channels A, B. The encoder and decoder networks can be set to any form of convolutional neural network. Interestingly, to avoid the uncertainty brought by traditional generative methods such as VAE, the author transforms the generation task into a classification one. The first figure out the common locating area of (A, B) and then split it into 313 categories. The classification is performed through a softmax layer with hyperparameter T as an adjustment. Based on [147], a range of colorization-based representation methods [69], [70], [148] are proposed to benefit downstream tasks.

Inpainting [55], [89] is more straight forward. We will ask the model to predict an arbitrary part of an image given the rest of it. Then a discriminator is employed to distinguish the inpainted image from the original one. Super-resolution method SRGAN [72] follows the same idea to recover high-resolution images from blurred low-resolution ones in the adversarial setting.

5.3 Pre-Trained Language Model

For a long time, the pre-trained language model (PTM) focuses on maximum likelihood estimation based pretext task because discriminative objectives are thought to be helpless due to languages' vibrant patterns. However, recently some work shows excellent performance and sheds light on contrastive objectives' potential in PTM.

The pioneering work is ELECTRA [21], surpassing BERT given at the same computation budget. ELECTRA proposes Replaced Token Detection (RTD) and leverages GAN's structure to pre-train a language model. In this setting, the generator G is a small Masked Language Model (MLM), which replaces masked tokens in a sentence to words. The discriminator D is asked to predict which words are replaced. Notice that *replaced* means not the same with original unmasked inputs. The training is conducted in two stages:

1) Warm-up the generator: train the G with MLM pretext task $\mathcal{L}_{\mathrm{MLM}}(\S, \theta_{\mathrm{G}})$ for some steps to warm up the parameters.
2) Trained with the discriminator: D's parameters is initialized with G's and then trained with the discriminative objective $\mathcal{L}_{\mathrm{D}\rangle f\rfloor}(\S, \theta_{\mathrm{D}})$ (a cross-entropy loss). During this period, G's parameter is frozen.

The final objective could be written as

$$\min_{\theta_{\mathrm{G}}, \theta_{\mathrm{D}}} \sum_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}_{\mathrm{MLM}}(\S, \theta_{\mathrm{G}}) + \lambda \mathcal{L}_{\mathrm{D}\rangle f\rfloor}(\S, \theta_{\mathrm{D}}). \quad (19)$$

Though ELECTRA is structured as GAN, it is not trained in the GAN setting. Compared to image data, which is continuous, word tokens are discrete, which stops the gradient backpropagation. A possible substitution is to leverage policy gradient, but ELECTRA experiments show that performance is slightly lower. Theoretically speaking, $\mathcal{L}_{\mathrm{D}\rangle f\rfloor}(\S, \theta_{\mathrm{D}})$ is actually turning the conventional k-class softmax classification into a binary classification. This substantially saves the computation effort but may somehow harm the representation quality due to the early degeneration of embedding space. In summary, ELECTRA is still an inspiring pioneer work in leveraging discriminative objectives.

At the same time, WKLM [134] proposes to perform RTD at the entity-level. For entities in Wikipedia paragraphs, WKLM replaced them with similar entities and trained the language model to distinguish them in a similar discriminative objective as ELECTRA, performing exceptionally well in downstream tasks like question answering. Similar work is REALM [45], which conducts higher article-level retrieval augmentation to the language model. However, REALM is not using the discriminative objective.

5.4 Graph Learning

There are also attempts to utilize adversarial learning ([23], [28], [128]). Interestingly, their ideas are quite different from each other.

The most natural idea is to follow BiGAN [33] and ALI [36]'s a practice that asks discriminator to distinguish representation from generated and prior distribution. Adversarial Network Embedding (ANE) [23] designs a generator G that is updated in two stages: 1) G encodes sampled graph into target embedding and computes traditional NCE with a context encoder F like Skip-gram, 2) discriminator D is asked to distinguish embedding from G and a sampled one from a prior distribution. The optimized objective is a sum of the above two objectives, and the generator G could yield better node representation for the classification task.

GraphGAN [128] considers to model the link prediction task and follow the original GAN style discriminative objective to distinguish directly at node-level rather than representation-level. The model first selects nodes from the target

node's subgraph v_c according to embedding encoded by the generator G. Then some neighbor nodes to v_c selected from the subgraph, together with those selected by G, are put into a binary classifier D to decide whether they are linked to v_c. Because this framework involves a discrete selection procedure, while gradient descents could update the discriminator D, the generator G is updated via policy gradients.

GraphSGAN [28] applies the adversarial method in semi-supervised graph learning with the motivation that marginal nodes cause most classification errors in the graph. Consider samples in the same category; they are usually *clustered* in the embedding space. Between clusters, there are *density gaps* where few samples exist. The author provides rigorous mathematical proof that we can perform complete classification theoretically if we generate enough fake samples in density gaps. GraphSGAN leverages a generator G to generate fake nodes in density gaps during the training and asks the discriminator D to classify nodes into their original categories and a category for those fake ones. In the test period, fake samples are removed, and classification results on original categories could be improved substantially.

5.5 Domain Adaptation and Multi-Modality Representation

Essentially, the discriminator in adversarial learning serves to match the discrepancy between latent representation distribution and data distribution. This function naturally relates to domain adaptation and multi-modality representation problems, aiming to align different representation distribution. [1], [2], [37], [105] studies how GAN can help on domain adaptation. [12], [129] leverage adversarial sampling to improve the negative samples' quality. For multi-modality representation, [151]'s image to image translation, [106]'s text style transfer, [22]'s word to word translation and [104] image to text translation show great power of adversarial representation learning.

5.6 Pros and Cons

Generative-contrastive (adversarial) self-supervised learning is particularly successful in image generation, transformation and manipulation, but there are also some challenges for its future development:

- *Limited applications in NLP and graph.* Due to the discrete nature of languages and graphs, the adversarial methods do not perform as well as they do in computer vision. Furthermore, GAN-based language generation has been found to be much worse than unidirectional language models such as GPTs.
- *Easy to collapse.* It is also notorious that adversarial models are prone to collapse during the training, with numerous techniques developed to stabilize its training, such as spectral normalization [83], W-GAN [3] and so on.
- *Not for feature extraction.* Although works such as BiGAN [33] and BigBiGAN [34] have explored some ways to leverage GAN's learned latent representation and achieve good performance, contrastive

learning has soon outperformed them with fewer parameters.

Despite the challenges, however, it is still promising because it overcomes some inherent deficits of the point-wise generative objective. Maybe we still need to wait for a better future implementation of this idea.

6 DISCUSSIONS AND FUTURE DIRECTIONS

In this section, we will discuss several open problems and future directions in self-supervised learning for representation.

Theoretical Foundation. Though self-supervised learning has achieved great success, few works investigate the mechanisms behind it. In this survey, we have listed several recent works on this topic and show that theoretical analysis is significant to avoid misleading empirical conclusions.

In [4], researchers present a conceptual framework to analyze the contrastive objective's function in generalization ability. [120] empirically proves that mutual information is only loosely related to the success of several MI-based methods, in which the sampling strategies and architecture design may count more. This type of works is crucial for self-supervised learning to form a solid foundation, and more work related to theory analysis is urgently needed.

Transferring to Downstream Tasks. There is an essential gap between pre-training and downstream tasks. Researchers design elaborate pretext tasks to help models learn some critical features of the dataset that can transfer to other jobs, but sometimes this may fail to realize. Besides, the process of selecting pretext tasks seems to be too heuristic and tricky without patterns to follow.

A typical example is the selection of pre-training tasks in BERT and ALBERT. BERT uses Next Sentence Prediction (NSP) to enhance its ability for sentence-level understanding. However, ALBERT shows that NSP equals a naive topic model, which is far too easy for language model pre-training and even decreases BERT's performance.

For the pre-training task selection problem, a probably exciting direction would be to design pre-training tasks for a specific downstream task automatically, just as what Neural Architecture Search [154] does for neural network architecture.

Transferring Across Datasets. This problem is also known as how to learn inductive biases or inductive learning. Traditionally, we split a dataset into the training used for learning the model parameters and the testing part for evaluation. An essential prerequisite for this learning paradigm is that data in the real world conform to our dataset's distribution. Nevertheless, this assumption frequently fails in experiments.

Self-supervised representation learning solves part of this problem, especially in the field of natural language processing. Vast amounts of corpora used in the language model pre-training help cover most language patterns and, therefore, contribute to the success of PTMs in various language tasks. However, this is based on the fact that text in the same language shares the same embedding space. For other tasks like machine translation and fields like graph learning where embedding spaces are different for different

datasets, learning the transferable inductive biases efficiently is still an open problem.

Exploring Potential of Sampling Strategies. In [120], the authors attribute one of the reasons for the success of mutual information-based methods to better sampling strategies. MoCo [47], SimCLR [15], and a series of other contrastive methods may also support this conclusion. They propose to leverage super large amounts of negative samples and augmented positive samples, whose effects are studied in deep metric learning. How to further release the power of sampling is still an unsolved and attractive problem.

Early Degeneration for Contrastive Learning. Contrastive learning methods such as MoCo [47] and SimCLR [15] are rapidly approaching the performance of supervised learning for computer vision. However, their incredible performances are generally limited to the classification problem. Meanwhile, the generative-contrastive method ELETRA [21] for language model pre-training is also outperforming other generative methods on several standard NLP benchmarks with fewer model parameters. However, some remarks indicate that ELETRA's performance on language generation and neural entity extraction is not up to expectations.

Problems above are probably because the contrastive objectives often get trapped into embedding spaces' early degeneration problem, which means that the model overfits to the discriminative pretext task too early, and therefore lost the ability to generalize. We expect that there would be techniques or new paradigms to solve the early degeneration problem while preserving contrastive learning's advantages.

7 CONCLUSION

This survey comprehensively reviews the existing self-supervised representation learning approaches in natural language processing (NLP), computer vision (CV), graph learning, and beyond. Self-supervised learning is the present and future of deep learning due to its supreme ability to utilize Web-scale unlabeled data to train feature extractors and context generators efficiently. Despite the diversity of algorithms, we categorize all self-supervised methods into three classes: generative, contrastive, and generative contrastive according to their essential training objectives. We introduce typical and representative methods in each category and sub-categories. Moreover, we discuss the pros and cons of each category and their unique application scenarios. Finally, fundamental problems and future directions of self-supervised learning are listed.

REFERENCES

[1] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," 2014, *arXiv:1412.4446*.

[2] F. Alam, S. Joty, and M. Imran, "Domain adaptation with adversarial training and graph embeddings," 2018, *arXiv:1805.05151*.

[3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[4] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, "A theoretical analysis of contrastive unsupervised representation learning," 2019, *arXiv:1902.09229*.

[5] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to retrieve reasoning paths over wikipedia graph for question answering," 2019, *arXiv:1911.10470*.

[6] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 15509–15519.

[7] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang, "SimGNN: A neural network approach to fast graph similarity computation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 384–392.

[8] D. H. Ballard, "Modular learning in neural networks," in *Proc. 6th Nat. Conf. Artif. Intell.*, 1987, pp. 279–284.

[9] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 2017.

[11] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*.

[12] L. Cai and W. Y. Wang, "KBGAN: Adversarial learning for knowledge graph embeddings," 2017, *arXiv:1711.04071*.

[13] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[14] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," 2020, *arXiv:2006.09882*.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv:2002.05709*.

[16] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," 2020, *arXiv:2006.10029*.

[17] T. Chen, Y. Sun, Y. Shi, and L. Hong, "On sampling strategies for neural network-based collaborative filtering," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 767–776.

[18] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[19] X. Chen and K. He, "Exploring simple siamese representation learning," 2020, *arXiv:2011.10566*.

[20] L. Chongxuan, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4088–4098.

[21] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," 2020, *arXiv:2003.10555*.

[22] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," 2017, *arXiv:1710.04087*.

[23] Q. Dai, Q. Li, J. Tang, and D. Wang, "Adversarial network embedding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2167–2174.

[24] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2978–2988.

[25] V. R. de Sa, "Learning classification with unlabeled data," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.*, 1994, pp. 112–119.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[28] M. Ding, J. Tang, and J. Zhang, "Semi-supervised learning on graphs with generative adversarial nets," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 913–922.

[29] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, "Cognitive graph for multi-hop reading comprehension at scale," 2019, *arXiv:1905.05460*.

[30] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," 2014, *arXiv:1410.8516*.

[31] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," 2016, *arXiv:1605.08803*.

[32] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1422–1430.

[33] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2016, *arXiv:1605.09782*.

[34] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 10541–10551.

[35] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1320–1329.

[36] V. Dumoulin et al., "Adversarially learned inference," 2016, *arXiv:1606.00704*.

[37] Y. Ganin et al., "Domain-adversarial training of neural networks," *The J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[38] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, *arXiv:1803.07728*.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[40] I. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[41] P. Goyal et al., "Self-supervised pretraining of visual features in the wild," 2021, *arXiv:2103.01988*.

[42] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.

[43] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[44] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 297–304.

[45] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," 2020, *arXiv:2002.08909*.

[46] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," 2020, *arXiv:2006.05582*.

[47] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019, *arXiv:1911.05722*.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[49] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," 2018, *arXiv:1808.06670*.

[50] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 2722–2730.

[51] W. Hu et al., "Strategies for pre-training graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2019.

[52] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "GPT-GNN: Generative pre-training of graph neural networks," 2020, *arXiv:2006.15437*.

[53] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," 2020, *arXiv:2003.01332*.

[54] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[55] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, 2017.

[56] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[57] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," 2019, *arXiv:1902.06162*.

[58] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, 2020.

[59] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," 2020, *arXiv:2004.04906*.

[60] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[61] D. Kim, D. Cho, D. Yoo, and I. S. Kweon, "Learning image representations by completing damaged jigsaw puzzles," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 793–802.

[62] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 10215–10224.

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

[64] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[65] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.

[66] L. Kong, C. D. M. d'Autume, W. Ling, L. Yu, Z. Dai, and D. Yogatama, "A mutual information maximization perspective of language representation learning," 2019, *arXiv:1910.08350*.

[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[68] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite bert for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[69] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 577–593.

[70] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6874–6883.

[71] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nat.*, vol. 521, no. 7553, pp. 436–444, 2015.

[72] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[73] D. Li, W.-C. Hung, J.-B. Huang, S. Wang, N. Ahuja, and M.-H. Yang, "Unsupervised visual representation learning by graph-based consistent constraints," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 678–694.

[74] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lectures Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[75] Y. Liu et al., "RoBERTa: A robustly optimized bert pretraining approach," 2019, *arXiv:1907.11692*.

[76] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[77] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*.

[78] M. Mathieu, "Masked autoencoder for distribution estimation," 2015.

[79] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, 2013.

[80] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[81] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," 2019, *arXiv:1912.01991*.

[82] J. Mitrovic, B. McWilliams, J. Walker, L. Buesing, and C. Blundell, "Representation learning via invariant causal mechanisms," 2020, *arXiv:2010.07922*.

[83] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.

[84] A. Newell and J. Deng, "How useful is self-supervised pretraining for visual tasks?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7345–7354.

[85] A. Ng et al., "Sparse autoencoder," *CS294A Lecture Notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[86] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[87] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9359–9367.

[88] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[89] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.

[90] Z. Peng, Y. Dong, M. Luo, X. M. Wu, and Q. Zheng, "Self-supervised graph representation learning via global context prediction," 2020, *arXiv:2003.01604*.

[91] Z. Peng, Y. Dong, M. Luo, X.-M. Wu, and Q. Zheng, "Self-supervised graph representation learning via global context prediction," 2020, *arXiv:2003.01604*.

[92] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[93] M. Popova, M. Shvets, J. Oliva, and O. Isayev, "MolecularRNN: Generating realistic molecular graphs with optimized properties," 2019, *arXiv:1905.13372*.

[94] J. Qiu et al., "GCC: Graph contrastive coding for graph neural network pre-training," 2020, *arXiv:2006.09963*.

[95] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "DeepInf: Social influence prediction with deep learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2110–2119.

[96] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," 2020, *arXiv:2003.08271*.

[97] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[98] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training."

[99] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019, Art. no. 9.

[100] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," 2016, *arXiv:1606.05250*.

[101] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 14837–14847.

[102] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," 2019, *arXiv:1908.10084*.

[103] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 385–394.

[104] N. Sarafianos, X. Xu, and I. A. Kakadiaris, "Adversarial representation learning for text-to-image matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5814–5824.

[105] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Adversarial representation learning for domain adaptation," *Stat*, vol. 1050, 2017, Art. no. 5.

[106] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6830–6841.

[107] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, and J. Tang, "GraphAF: A flow-based autoregressive model for molecular graph generation," 2020, *arXiv:2001.09382*.

[108] A. Sinha et al., "An overview of microsoft academic service (MAS) and applications," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 243–246.

[109] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," Dept. Comput. Sci., Colorado Univ. Boulder, Boulder, CO, 1986.

[110] F.-Y. Sun, J. Hoffmann, and J. Tang, "InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," 2019, *arXiv:1908.01000*.

[111] F.-Y. Sun, M. Qu, J. Hoffmann, C.-W. Huang, and J. Tang, "vGraph: A generative model for joint community detection and node representation learning," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 512–522.

[112] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, pp. 5892–5899.

[113] K. Sun, Z. Zhu, and Z. Lin, "Multi-stage self-supervised learning for graph convolutional networks," 2019, *arXiv:1902.11038*.

[114] Y. Sun et al., "ERNIE: Enhanced representation through knowledge integration," 2019, *arXiv:1904.09223*.

[115] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[116] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 990–998.

[117] W. L. Taylor, ""Cloze procedure": A new tool for measuring readability," *Journalism Quart.*, vol. 30, no. 4, pp. 415–433, 1953.

[118] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2019, *arXiv:1906.05849*.

[119] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning," 2020, *arXiv:2005.10243*.

[120] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, "On mutual information maximization for representation learning," 2019, *arXiv:1907.13625*.

[121] A. van den Oord et al., "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synth. Workshop*, 2016, pp. 125–125.

[122] A. Van den Oord et al., "Conditional image generation with pixelcnn decoders," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4790–4798.

[123] A. van den Oord et al., "Neural discrete representation learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6306–6315.

[124] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1747–1756.

[125] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[126] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[127] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," 2018, *arXiv:1809.10341*.

[128] H. Wang et al., "GraphGAN: Graph representation learning with generative adversarial networks," 2018, pp. 2508–2515.

[129] P. Wang, S. Li, and R. Pan, "Incorporating GAN for negative sampling in knowledge representation learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2005–2012.

[130] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks: A survey and taxonomy," 2019, *arXiv:1906.01529*.

[131] C. Wei et al., "Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1910–1919.

[132] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.

[133] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10687–10698.

[134] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, "Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model," 2019, *arXiv:1912.09637*.

[135] K. Xu, J. Li, M. Zhang, S. S. Du, K.-I. Kawarabayashi, and S. Jegelka, "How neural networks extrapolate: From feedforward to graph neural networks," 2020, *arXiv:2009.11848*.

[136] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram, and D. Mahajan, "ClusterFit: Improving generalization of visual representations," 2019, *arXiv:1912.03330*.

[137] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5147–5156.

[138] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5754–5764.

[139] Z. Yang et al., "HotpotQA: A dataset for diverse, explainable multi-hop question answering," 2018, *arXiv:1809.09600*.

[140] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 6410–6421.

[141] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "GraphRNN: Generating realistic graphs with deep auto-regressive models," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5708–5717.

[142] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," 2020, *arXiv:2010.13902*.

[143] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?" 2020, *arXiv:2006.09136*.

[144] F. Zhang *et al.*, "OAG: Toward linking large-scale heterogeneous entity graphs," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2585–2595.

[145] J. Zhang, Y. Dong, Y. Wang, J. Tang, and M. Ding, "ProNE: Fast and scalable network representation learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4278–4284.

[146] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4438–4445.

[147] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 649–666.

[148] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1058–1067.

[149] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," 2019, *arXiv:1905.07129*.

[150] D. Zhu, P. Cui, D. Wang, and W. Zhu, "Deep variational network embedding in wasserstein space," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2827–2836.

[151] J.-Y. Zhu *et al.*, "Toward multimodal image-to-image translation," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 465–476.

[152] C. Zhuang, A. L. Zhai, and D. Yamins, "Local aggregation for unsupervised learning of visual embeddings," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6002–6012.

[153] B. Zoph *et al.*, "Rethinking pre-training and self-training," 2020, *arXiv:2006.06882*.

[154] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*.

Xiao Liu is currently working toward the senior undergraduate degree with the Department of Computer Science and Technology, Tsinghua University, China. His main research interests include data mining, machine learning and knowledge graph. He has published a paper on KDD.

Fanjin Zhang received the bachelor's degree from the Department of Computer Science and Technology, Nanjing Unviersity, China. She is currently working toward the PhD degree with the Department of Computer Science and Technology, Tsinghua University, China. Her research interests include data mining and social network.

Zhenyu Hou is currently working toward the undergraduate degree with the Department of Computer Science and Technology, Tsinghua University, China. His main research interests include graph representation learning and reasoning.

Li Mian received the bachelor's degree from the Department of Computer Science, Beijing Institute of Technology, China, in 2020. She is now admitted into a graduate program in Georgia Institute of Technology, Atlanta, Georgia. Her research interests focus on data mining, natural language processing and machine learning.

Zhaoyu Wang is currently working toward the graduate degree with the Department of Computer Science and Technology, Anhui University, China. His research interests include data mining, natural language processing and their applications in recommender systems.

Jing Zhang received the master's and PhD degrees from the Department of Computer Science and Technology, Tsinghua University, China. She is an assistant professor in Information School, Renmin University of China, China. Her research interests include social network mining and deep learning.
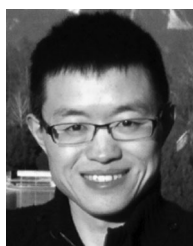
Jie Tang (Fellow, IEEE) received the PhD degree from Tsinghua University, China. He is full professor with the Department of Computer Science and Technology, Tsinghua University, China. His main research interests include data mining, social network, and machine learning. He has published more than 200 research papers in top international journals and conferences.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.