GenURL: A General Framework for Unsupervised Representation Learning

Siyuan Li, Graduate Student Member, IEEE, Zicheng Liu, Student Member, IEEE,
Zelin Zang, Graduate Student Member, IEEE, Di Wu, Zhiyuan Chen,
and Stan Z. Li, Fellow, IEEE

Abstract—Unsupervised representation learning (URL) that learns compact embeddings of high-dimensional data without supervision has achieved remarkable progress recently. However, the development of URLs for different requirements is independent, which limits the generalization of the algorithms, especially prohibitive as the number of tasks grows. For example, dimension reduction (DR) methods, t-SNE and UMAP, optimize pairwise data relationships by preserving the global geometric structure, while self-supervised learning, SimCLR and BYOL, focuses on mining the local statistics of instances under specific augmentations. To address this dilemma, we summarize and propose a unified similarity-based URL framework, GenURL, which can adapt to various URL tasks smoothly. In this article, we regard URL tasks as different implicit constraints on the data geometric structure that help to seek optimal low-dimensional representations that boil down to data structural modeling (DSM) and low-dimensional transformation (LDT). Specifically, DSM provides a structure-based submodule to describe the global structures, and LDT learns compact low-dimensional embeddings with given pretext tasks. Moreover, an objective function, general Kullback–Leibler (GKL) divergence, is proposed to connect DSM and LDT naturally. Comprehensive experiments demonstrate that GenURL achieves consistent state-of-the-art performance in self-supervised visual learning, unsupervised knowledge distillation (KD), graph embeddings (GEs), and DR.

Index Terms—Contrastive learning (CL), dimension reduction (DR), graph embedding (GE), knowledge distillation (KD), self-supervised learning.

I. INTRODUCTION

Learning low-dimensional representations from complex data without human supervision, i.e., unsupervised representation learning (URL), is a long-standing problem. As the high-dimensional data are usually highly redundant and non-Euclidean, a widespread assumption is that data lie in a low-dimensional ambient space. However, URL algorithms under different tasks and data structures are designed independently of each other, yet they have the same ultimate goal of finding the desired embedding space.

URL studies are now broadly categorized into three types of applications: 1) dimension reduction (DR) and graph embedding (GE) algorithms aim to encode non-Euclidean input data into a latent space Z plainly without any prior knowledge of the related domains, as shown in Fig. 1 (left and middle); 2) complementary to DR and GE, another popular path of URL focuses on data-specific augmentations, such as image crop, which leads to a clustering structure and learns discriminative representations, as illustrated in Fig. 1 (right); and 3) in addition, knowledge distillation (KD) is another approach that can be regarded as an implicit URL method transferring the knowledge from the pretrained model to enhance the student model unsupervised instead of considering geometric structure or special prior information of the target data. Specifically, from the perspective of algorithmic bias, these representative URL algorithms are grouped into two classes: global structure-oriented, e.g., t-SNE and UMAP, and individual augmentation-oriented, e.g., SimCLR and MoCo, respectively. It is a fact that these two independent algorithms are designed to excel in their respective areas of applicability. There is, thus, a natural question if the intrinsic representation of data is determined by both the global data structure and data-specific prior assumptions in a unified framework.

This work: a general framework of URL. Based on the above URL methods, developing an effective and unified URL framework adaptive to various scenarios is a new trend in the community [5], [6]. In this work, we consider both

![Fig. 1. Illustration of various empirical structures of high-dimensional data.](image-url)

We encode COIL-20 [1], CiteSeer [2], and STL-10 [3] to 2-dim, 128-dim, and 512-dim by GenURL (128-dim and 512-dim latent spaces are then visualized by UMAP [4] in 2-dim). Left: we preserve local geometric structures of the circle manifolds in COIL-20 in the DR task. Middle: topological and geometric structures of citation networks in CiteSeer are encoded in the GE task. Right: with the instance discriminative proxy task, we learn a discriminative representation in the validation dataset of STL-10.
the global structure and local discriminative statistics and, thus, reformulate the URL problem as a non-Euclidean data embedding problem that encodes the structure and content parallelly in a compact low-dimensional space. For instance, we introduce an effective and general framework of URL called GenURL, containing two steps: data structural modeling (DSM) and low-dimensional transformation (LDT), as shown in Fig. 2. To model the global structures, we combine the (DSM) and low-dimensional transformation (LDT), as shown in Fig. 2. To model the global structures, we combine the

A. Dimension Reduction

Adopting the manifold assumption in DR, which assumes data lie on a low-dimensional manifold immersed in the high-dimensional space, most DR methods try to preserve intrinsic geometric properties of data [7], [8], [9], [10], [11], [12]. Another practical branch of DR introduced by t-SNE [13] and UMAP [4] optimizes the pairwise similarities between latent and input spaces. More recently, deep manifold learning methods can learn more complex manifolds and can be transferred to unseen data by using neural networks. Parametric t-SNE (P-SNE) [14], parametric UMAP (P-UMAP) [15], DMT [16], and its variant [17] are proposed directly based on t-SNE and UMAP. However, current DR methods model desired data structures only relying on the geometry of input space and might fail with highly redundant data, such as natural images.

B. Graph Embedding

GE transfers graph data into a low-dimensional and continuous feature space while preserving most graph structures and topological and geometric structures, such as vertex content. Most early methods are model-free, which contains four categories: Laplacian eigenmaps-based [18], local similarity-based [19], [20], [21], matrix factorization-based [22], and nonparametric Bayesian modeling-based methods [23]. Recently, model-based methods [24], [25] utilize graph convolutional networks (GCNs) [26] or graph autoencoders [27], [28] to learn both graph structures and feature information. More recently, some methods [29] take both the geometric and topological structures into consideration. This type of learning, which relies exclusively on the graph structure, eventually leads to the problem of homogenization of representations, so in addition to conveying information through the structure, the nodes themselves need to be bounded by independent a priori information.

C. Self-Supervised Visual Representation Learning

Early SSL methods design handcrafted pretext tasks [30], [31], [32], [33], which rely on somewhat ad hoc heuristics and have limited abilities to capture practically useful representations. Another popular form is clustering-based methods [34], [35], [36], [37], learning discriminative representation by offline or online clustering. Recently, contrastive learning (CL) [38], [39], [40], [41], which discriminates positive pairs against negative pairs, achieved state-of-the-art performance

Fig. 2. Illustration of GenURL. The data structures are first modeled as similarity $P_X$ by calculating the graph distance on each predefined graph. Then, the LDT mapping $f_0$ is optimized by minimizing $L$ based on the fixed $P_X$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
in various vision tasks. Different mechanisms [41], [42], [43], [44] are proposed to prevent trivial solutions in CL to learn useful representations. To fully utilize negative samples, Chuang et al. [45], Kalantidis et al. [46], Robinson et al. [47], and Li et al. [48] explore hard samples in the momentum memory bank. Meanwhile, some efforts have been made on top of contrastive methods to improve pretraining quality for specific downstream tasks [49], [50], [51], [52]. More recently, with the introduction of the vision Transformers [53], [54], masked image modeling (MIM) [55], [56], [57], [58] achieved state-of-the-art performance based on Transformers, which randomly mask out patches in the input image and predict the masked patches with decoder. While SSL can extract highly redundant data features, the loss of geometric structure constraints is an inability to model a priori semantic relationships between clusters in a global context, such as temporal evolutionary order.

D. Knowledge Distillation

KD was first proposed by Hinton et al. [59], which aims to transfer knowledge from trained neural networks to a smaller one without losing too much generalization power. There are three types of existing KD methods: response-based [59], [60], [61] and feature-based [62] methods require labels to utilize intermediate-level supervision from the teacher model. Relation-based KD [63], [64], [65], [66], [67] methods explore the relationships between different layers or data sample pairs, which can work without supervision and extend to self-supervised settings [67], [68], [69]. However, the performance of replicating the teacher model alone is unsatisfactory, and considering both the data structure and the a priori assumptions is a critical step in improving the efficiency of transfer learning. Our method handles the KD task as a special type of DR task without label supervision.

We summarize and analyze the above URL approaches and propose GenURL, a framework that successfully links the two important elements of URLs, data geometry and task hypotheses, based on generalized similarity. GenURL not only takes into account the data geometries that are the focus of DE and GE but also introduces task-relevant data and model hypotheses from CL and KD to enhance the overall quality of the representations and, thus, improve the performance of the downstream tasks.

III. METHOD
A. Preliminaries

The general goal of URL is identical across various URL tasks: Given a finite set of samples $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^{n \times D}$, we seek a continuous mapping, $f_0(x): \mathbb{R}^D \rightarrow \mathbb{R}^d$, where $d \ll D$, that transfers data into a compact latent space $Z$ while preserving essential structures of $X$ to facilitate most downstream tasks [25], [70], [71]. In general, the intrinsic structures of data are determined by both the input data and task-specific assumptions. However, most research in URL has focused on individual data modalities or specific tasks, resulting in specific designs and different learning objectives. We take the following four widely used URL tasks as examples.

D. Knowledge Distillation

Although there are various task-specific assumptions in URL tasks, we may summarize them as a data embedding problem based on the manifold assumption. Assuming the data $X$ are constrained on a compact low-dimensional manifold $M$, a neighborhood system for each sample $x_i$ is defined as $N_i \subset \mathbb{R}^d$, and $M$ is supported on the mixture of these neighborhood systems, $M = \cup_{i=1}^n N_i$. The discriminative property of $N$ facilitates various downstream tasks. Since different neighborhood systems are disjoint, we use an adjacency matrix $A$ to represent neighborhood systems: $A_{ij} = 1$ indicates $x_i$ and $x_j$ are in the same neighborhood system, or $A_{ij} = 0$. $A$ can be built by an undirected graph $G = (X, E)$ provided by a specific URL task (discussed in Section IV). Local (neighborhood systems) and global (manifold) structures are two essential properties for learning good representations: local geometries describe the discriminative features of instances, while relationships between neighborhood systems reflect the global view of topological structures. Based on $M$, the similarity between two nonadjacent samples within each $N_i$ can be approximated by the shortest-path distance. Since most URL algorithms are designed for specific tasks or data, the following two typical issues arise.

1) Over-Uniformity: Without the global structure, the optimal solution for constructing $N$ is to place each $x$ evenly in the embedding space, as shown in the middle of Fig. 3. In other words, the decision boundaries are maximized to the discrimination between $x$. However, the manifold structure is then damaged and unorganized in this case, which means the learned representations cannot be generalized to other downstream tasks [72], such as data visualization.

2) Ill-Clustering: The converse is also true; if we focus excessively on the global structure and ignore instance differences, a local collapse will occur. The result is shown on the right of Fig. 3, also called local homogenization. A classical case in the node classification task of graph data is the over-smoothing issue [73]: global-based message passing makes the connected nodes ultimately nondistinguishable.

Therefore, there is a question worth thinking about whether we can solve both of these issues at the same time in a mutually constraining way. Motivated by this, we propose a general and unified framework for URL that can be adapted to various URL tasks effectively.

B. Similarity-Preserving Framework

Given a set of $m$ empirical graphs $G = \{G_t\}_{t=1}^m$ defined on the data $X$, where $G_t = (X, d_t)$, we calculate the pairwise
distance $d_X$ based on $X$ and $G_t$ to model the empirical data structures. To capture the local geometry defined in $G_t$ and eliminate the scale factor between different distances, we adopt the similarly defined in $[0, 1]$ by converting the pairwise distance $d$ to the similarity with a long-tailed $t$-distribution kernel function $\kappa(.)$

$$\kappa(d, v) = \sqrt{2\pi} \cdot \Gamma\left(\frac{v+1}{2}\right) \left(1 + \frac{d^2}{v}\right)^{-\frac{v+1}{2}}$$

(1)

where $v$ denotes the degree of $t$-distribution freedom. Notice that the $t$-distribution approximates the Gaussian distribution when $v \rightarrow +\infty$ and approximates the uniform distribution when $v \rightarrow 0$. The latent space is usually a (normalized) Euclidean space $(Z, d_Z)$. The similarities of input and latent spaces are written as follows:

$$p_X(x_i, x_j) = \alpha_i \sum_{i=1}^{m} A_{ij} \kappa(d_X(x_i, x_j), \nu_X)$$

(2)

$$p_Z(z_i, z_j) = \kappa(d_Z(z_i, z_j), \nu_Z)$$

(3)

where $\alpha_i$ is a balancing hyperparameter for $d_X$. Notice that $\sum_{i,j} p_X(x_i, x_j)$ and $\sum_{i,j} p_Z(z_i, z_j)$ are not equal to 1 in a mini-batch.

1) Data Structural Modeling: Since $p_X$ can reflect the reliability of relations between $x$ and other samples, we can control learned representations by the $push$ and $pull$ forces with various $\nu_X$ and $\nu_Z$. Taking the DR task as an example, we assume $d_X$ is reliable in the original structure of input data, while $d_Z$ is likely to be distorted in the extremely low-dimensional space (e.g., 2-dim), as shown in Fig. 4: We set $\nu_X \rightarrow +\infty$ (i.e., the Gaussian distribution) giving the local sample pair $(x_i, x_j)$ and the disjoint sample pair $(x_i, x_k)$ high and low similarities, respectively, while set $\nu_Z = 1$ to make $d_Z(x_i, x_j) \ll d_Z(x_i, x_k)$. As explained in Fig. 4, the $push$ and $pull$ forces enable the learned embedding preserving geometric and topological properties of the input data after optimizing (6). For practical purposes, we can fix $\nu_X$ and adjust $\nu_Z$ in $[100, 0]$ to control the structure of the latent space based on the characters of URL tasks (detailed in Section V-F).

We provide static and dynamic methods to adaptively model the similarity $p_X$ based on (2). As for the static, we first normalize $d_X$, as $d'_X(x_i, x_j) = ((d_X(x_i, x_j) - \mu_{i,j})/\sigma_i)$, where $\mu_{i,j}$ measures the distance scale of each $x_i$ and $\sigma_i$ controls the extension of local neighborhood systems. We select proper $\mu_{i,j}$ and $\sigma_i$ in terms of population statistics of data (detailed in Section IV), such as mean and standard deviation of all samples. We rewrite (2) and (3) for the static $\tilde{p}_X$ as follows:

$$\tilde{p}_X(x_i, x_j) = \sum_{k=1}^{K} \alpha_k \kappa \left( \frac{d'_X(x_i, x_j) - \mu_{i,k}}{\sigma_k}, \nu_X \right)$$

(4)

$$\tilde{p}_Z(z_i, z_j) = \kappa \left( \frac{d'_Z(z_i, z_j) - \mu_{Z}}{\sigma_Z}, \nu_Z \right)$$

(5)

Then, we embed $X$ into the latent space $Z$ by minimizing the dissimilarity between $\tilde{p}_X$ and $\tilde{p}_Z$ by a loss function $\mathcal{L}(.)$

$$\min_{\theta} \mathcal{L}(\tilde{p}_X, \tilde{p}_Z).$$

(6)

2) Low-Dimensional Transformation: Notice that the static $\tilde{p}_X$ mainly relies on the balancing weight $\alpha_i$ for each $d_X$, resulting in suboptimal performances when some distances are unreliable. Thus, we design the dynamic method utilized in the early stage of the encoder $f_0$ to achieve reliable LDT, say, the $l$th layer where $l \in [1, L-1]$. Since the encoder $f_0$ will gradually capture data structures by optimizing (6), we regard the latent space of the $l$th stage $\tilde{p}_Z(l)$ as the dynamic $\tilde{p}_X$, which adaptively combines various $d_X$. Based on the static $\tilde{p}_X$, we define the dynamic $\tilde{p}'_X = \beta \tilde{p}_X + \tilde{p}_Z(l)$, where $\beta$ is a weight, which linearly decays from 1 to 0. Notice that $\tilde{p}_Z(l)$ does not require gradient. Finally, our proposed GenURL is demonstrated in Fig. 2. As discussed in Sections IV and V, the static $\tilde{p}_X$ usually suits URL tasks with well-defined input spaces, such as DR and KD, while the dynamic fits other tasks, such as CL and GE.

C. Loss Function

As we formulate the URL problem as (6), where $\tilde{p}_X$ is regarded as the target, we discuss several similarity functions to achieve optimal embedding. Here, we consider $\tilde{p}_X$ in two cases: 1) generated by incomplete metric spaces where the relationship between distant neighbors is unknown and 2) generated by well-defined metric spaces. In case 1), we focus on preserving the structures of each neighborhood system (similar pairs). In case 2), we pay equal attention to dissimilar pairs to capture global relationships. We analyze these losses in various tasks by ablation studies in Section V.

1) Mean-Squared Error: Mean-squared error (MSE) is the most commonly used loss function to measure the similarity between $\tilde{p}_X$ and $\tilde{p}_Z$ with $L_2$-norm

$$\mathcal{L}_{MSE}(\tilde{p}_X, \tilde{p}_Z) = \sum_{i,j} [||\tilde{p}_X(x_i, x_j) - \tilde{p}_Z(z_i, z_j)||^2].$$

(7)

However, the MSE treats all sample pairs equally, resulting in suboptimal solutions in both cases.

2) Kullback–Leibler Divergence: The KL divergence is commonly used to measure the similarity between two probability distributions

$$\mathcal{L}_{KL}(\tilde{p}_X, \tilde{p}_Z) = -\sum_{i,j} \tilde{p}_X(x_i, x_j) \log \frac{\tilde{p}_X(x_i, x_j)}{\tilde{p}_Z(z_i, z_j)}.$$

(8)
When $\log \tilde{p}_X$ is constant, the KL divergence is equal to the cross entropy between $\tilde{p}_X$ and $\tilde{p}_Z$. We can regard $\tilde{p}_X$ as a reweight factor from similar sample pairs to dissimilar sample pairs. However, the KL divergence requires $\tilde{p}_Z(z_i, z_j) = 1$ [13]. When $\tilde{p}_Z$ is not a probability distribution [4], [44], it might result in a trivial solution, $\tilde{p}_Z(z_i, z_j) \to 0$ for each $\tilde{p}_Z(x_i, x_j)$.

3) Binary Cross Entropy: To make up the defect in the KL divergence, the binary cross-entropy (BCE) loss [4] adds a symmetric term for $p_X(x_i, x_j) \to 0$ (placing higher weights than the KL divergence) to prevent the trivial solution

$$L_{\text{BCE}}(\tilde{p}_X, \tilde{p}_Z) = -\sum_{i,j} \tilde{p}_X(x_i, x_j) \log \tilde{p}_Z(z_i, z_j)$$

$$-\sum_{i,j} (1 - \tilde{p}_X(x_i, x_j)) \log (1 - \tilde{p}_Z(z_i, z_j)).$$

(9)

The BCE loss optimizes the most similar and dissimilar pairs symmetrically, which is suitable to perverse a well-defined metric space in case 2).

4) GKL Divergence: Although the BCE loss can solve case 1) well under ideal conditions, it might suffer from outliers, e.g., false negative samples in SSL and GE tasks [29], [47], [74], resulting in performance degradation in case 2). Since we can regard the symmetric term in (9) as the normalization constrain, $\sum_{i,j} \tilde{p}_X(x_i, x_j) = \sum_{i,j} \tilde{p}_Z(z_i, z_j)$, it is a direct way to prevent the trivial solution in KL divergence. However, the symmetric term emphasizes the importance of the negative samples (dissimilar pairs) with the reweight factor $1 - \tilde{p}_X$. Therefore, we propose a relaxed version of the BCE loss with a relaxed symmetric term

$$L_{\text{GKL}}(\tilde{p}_X, \tilde{p}_Z) = -\sum_{i,j} \tilde{p}_X(x_i, x_j) \log \tilde{p}_Z(z_i, z_j)$$

$$+\gamma \sum_{i,j} ||\tilde{p}_X(x_i, x_j) - \tilde{p}_Z(z_i, z_j)||_p$$

(10)

where $\gamma$ is a balancing weight and $||.||_p$ is $L_p$-norm. We adopt $L_1$-norm and set $\gamma = 0.1$ in the GKL loss. Compared with the BCE, the GKL loss is less affected by unreliable samples (usually dissimilar samples) when some $d_{X,i}$ are not reliable.

IV. INSTANTIATION OF GENURL

GenURL generalizes different tasks by fully utilizing partial available information within corresponding datasets and patching the missing properties of URL modeling.

A. DR and GE

The goal of GE tasks is to encode the geometric and topological structures of the input data. Given a graph $G = (V, E, X)$, an adjacency matrix $A_1$ can be defined based on $G$ and another $A_2$ based on a kNN graph built with the feature space $X$. We calculate the shortest-path distance for the entire graph, $d_1(v_i, v_j) = ||v_i - v_j||_2$ when $A_{1,ij} = 1$, and set $d_1(v_i, v_j) \to \infty$ when $A_{1,ij} = 0$. The distance $d_2$ of the kNN graph is obtained in the same way. Intuitively, we define the input similarity $p_X \triangleq \alpha_1 p_1 + \alpha_2 p_2$, where $\alpha_1 = 1$. To remove the scale effects of distances in different spaces, we calculate $\mu_i = \min(d_i(x_i, x_0), \ldots, d_i(x_i, x_n))$ and $\sigma_i = (1/n) \sum_{i=1}^{n} \sigma_i$ in data $X$. Instead of GCN, used in most GE methods, we use five-layer MLP using leaky ReLU activation, with the middle latent dimension of 512 and the embedding dimension of 128. As for the DR task, it is regarded as a special case of GE tasks, which only requires a kNN graph. We adopt the static $\tilde{p}_X$ for both the tasks. Similarly, three-layer MLP is adopted for DR tasks with the 2-dim embedding space.

B. Self-Supervised Learning for Visual Representation

Unlike the DR and GE tasks, the distance on raw images in the open scenes is unreliable to reflect the desired low-dimensional structures for most discriminative downstream tasks, since most images are highly redundant and unstructured. Hence, we import the proxy knowledge of the instance discriminative task in CL [39], [40] as follows. Given a mini-batch of data $X^B = \{x_i\}_{i=1}^n$, we apply augmentation $\tau \sim T$ to each sample as $\tau(x_i)$ to obtain two correlated views $X^{b_i} = \{x^{b_i}\}_{i=1}^n$ and $X^{b_j} = \{x^{b_j}\}_{j=1}^n$, and fed to the encoder $f_{\theta}$ (e.g., ResNet [70]) and a projection MLP neck [40], denoted as $h_{\phi}$, producing batches of latent representations $Z^B_a, Z^B_b$ and $H^A_a, H^B_b$, where $z_i = f_{\theta}(x_i)$ and $h_i = h_{\phi}(z_i)$. The projection neck will be discarded after pretraining. We can convert the proxy knowledge of content invariance into an adjacency matrix $A$: $A_{ij} = 1$ for two different views $x_i$ and $x_j$. We adopt the $L_2$-normalized cosine distance of the projection as the latent representation, $d_Z \triangleq (h_i/||h_i||_2) \cdot (h_j/||h_j||_2)$. As we discussed in Section III-B, we adopt two versions of the input similarity. As for the static $\tilde{p}_X$, we use the discrete distance $d_1$ defined by the proxy knowledge and the Euclidean distance $d_2$ defined by kNN graph in $X$, $p_X \triangleq \alpha_1 p_1 + \alpha_2 p_2$, where $\alpha_1 = 1$ and $\alpha_2 = 0.01$, which is linearly decreased to 0. As for the dynamic, we calculate the cosine distance $(z_i/||z_i||_2) \cdot (z_j/||z_j||_2)$ and define $\tilde{p}_{Z_L}$ on the first latent space. The dynamic $\tilde{p}_X = \beta \tilde{p}_X + \tilde{p}_{Z_L}$.

C. Unsupervised KD

As for the KD task, we regard it as a special type of DR task, i.e., encode the compact latent space of pertained teachers $z^T$ into the lower latent space of a student. Since the input space is already a well-defined Euclidean metric space, where the distance of every sample pair can be measured by $d_2(x_i, x_j)$, i.e., the input distance $d_X \triangleq d_2$, we adopt the static method and use $L_2$-normalized cosine distance for the latent space of both the teacher and student. To fully explore the knowledge in teacher models, we should pay more attention to the global structural relation between distant samples while preserving the local geometry. As discussed in Section III-C, the BCE loss is more suitable for KD tasks.

V. EXPERIMENTS

In this section, we evaluate the effectiveness of GenURL on various unsupervised learning tasks, including self-supervised
visual representation (SSL), unsupervised KD, GE, and DR. Meanwhile, we conduct ablation studies of loss functions and hyperparameters to explore characters of various scenarios.

A. Experimental Setup

As for evaluation protocols, we adopt the linear protocol as the standard practice [39], [75], which trains a linear classifier on top of fixed representations. As for self-supervised visual representation, we further follow the semisupervised classification [40] and evaluate the generation ability of representations by transfer learning [39]. As for DR, we further adopt trustworthiness (Trust) and continuity (Cont) [11] to measure the distortion between the input data and representations. We use the following training settings for different tasks unless specified. We use Adam optimizer [76] with a learning rate of $lr = 0.0005$. The batch size is 256 by default. All experiments report the mean of three times as default. The best and second results are denoted by bold and underlined.

1) Datasets: Various types of datasets are used in diverse URL tasks. Image datasets include the following: 1) MNIST [78] contains gray-scale images of ten classes in $28 \times 28$ resolutions, 50k for training, and 10k for testing; 2) FashionMNIST [79] contains images of ten classes of 20 objects, for a total of 1440 images in $28 \times 28$ resolutions; 3) CIFAR-10/100 [80] contains 50k training images and 10k test images in $32 \times 32$ resolutions, with ten classes and 100 classes settings; 4) STL-10 [3] consists of 5k labeled training images for ten classes and 100k unlabeled training images and a test set of 8k images in $96 \times 96$ resolutions; 6) Tiny ImageNet (Tiny) [81] has 10k training images and 10k validation images of 200 classes in $64 \times 64$ resolutions; 7) ImageNet-1k (IN-1k) [82] contains 1.28M training images and 50k validation images from 1000 classes in $224 \times 224$ resolutions; datasets 1–3 are used for DR tasks, and 4–7 are used for SSL and KD tasks; 8) graph datasets for GE tasks include CORA [83] that contains binary word representations by transfer learning [39]. As for DR, we further adopt trustworthiness (Trust) and continuity (Cont) [11] to measure the distortion between the input data and representations.

2) Implementation of CL: We follow MoCo.v2 [84] for CL pretraining, which adopts ResNet [70] encoder with a two-layer MLP projector based on OpenMixup codebase [85]. All CL methods adopt the same network and augmentation settings, while other methods use default settings in their paper. The data augmentation setting in MoCo.v2 is as follows: geometric augmentations include RandomResizedCrop with the scale in $[0.2, 1.0]$ and RandomHorizontalFlip. Color augmentation includes ColorJitter with \{brightness, contrast, saturation, hue\} strength of $[0.4, 0.4, 0.4, 0.1]$ with an applying probability of 0.8, and RandomGrayscale with an applying probability of 0.2. Blurring augmentation uses a Gaussian kernel of size $23 \times 23$ with a standard deviation uniformly sampled in $[0.1, 2.0]$. As shown in Tables I and II, we use the GKL loss with $\gamma = \nu_x = \nu_Z = 100$ and $\sigma_x = \sigma_Z = 0.1$ for GE on CIFAR-10, CIFAR-100, STL-10, Tiny ImageNet, and ImageNet-1k datasets.

3) Implementation of KD: In KD tasks, GenURL follows the settings of the current-proposed contrastive-based KD method SEED [67], which adopts the nonlinear projector network and data augmentations used in MoCo.v2. Note that MoCo.v2 pretrained ResNet-50 is adopted as the teacher network and data augmentations used in MoCo.v2. Note that we use a binary search [requires $O(n^2)$] with five nearest neighbors for each data point, i.e., the optimal hyperparameters guarantee that the five nearest neighbors of $x_i$ have a large similarity score. There are similar practices in UMAP [4] and t-SNE [13]. If the dataset is too large, we set $\mu_2$ and $\sigma_2$ to the statistic mean and STD of the whole dataset.

4) Implementation of GE: In GE tasks, we adopt $L_2$ distance with $\sigma_Z = 1$ and tune various hyperparameters as follows. As for $\mu$ and $\sigma$, we perform a grid search of $\mu_Z$ and $\sigma_Z$ for the latent space in $[0.01, 0.1, 1, 10, 100]$ on the validation set. As for $\mu_2$ and $\sigma_2$ of the raw attribute space, we use a binary search [requires $O(n^2)$] with five nearest neighbors for each data point, i.e., the optimal hyperparameters guarantee that the five nearest neighbors of $x_i$ have a large similarity score. There are similar practices in UMAP [4] and t-SNE [13]. If the dataset is too large, we set $\mu_2$ and $\sigma_2$ to the statistic mean and STD of the whole dataset.

5) Implementation of DR: GenURL performs DR tasks with the BCE loss and the kNN graph built on the input X following

---

**TABLE I**

| method       | Linear  | Semi-supervised |
|--------------|---------|-----------------|
|              | Supervised |                  |
|              | 400ep     | 800ep           | 1600ep         | 400ep     | 800ep           | 1600ep         |
| Related loc [30] | 69.20 | 64.20           | 64.37          | 68.49 | 67.93           | 68.41          |
| Rotation [32] | 76.70 | 73.14           | 72.15          | 89.91 | 90.43           | 90.05          |
| NFID [38]    | 82.51 | 84.64           | 89.88          | 88.31 | 90.06           | 92.86          |
| ODC [35]     | 73.45 | 75.47           | 76.20          | 80.88 | 82.04           | 85.80          |
| SimCLR [40]  | 86.92 | 87.25           | 88.75          | 89.88 | 90.25           | 91.30          |
| MoCo.v2 [84] | 84.89 | 89.68           | 91.78          | 89.66 | 92.53           | 93.65          |
| BYOL [41]    | 81.17 | 88.74           | 91.41          | 85.38 | 91.71           | 92.69          |
| SwAV* [37]   | 84.35 | 88.79           | 91.02          | 86.77 | 92.05           | 92.63          |
| BarlowTwins [43] | 85.74 | 88.90           | 91.23          | 86.35 | 91.82           | 92.78          |
| GenURL       | **88.35** | **90.82** | **91.85**     | **90.88** | **92.58** | **93.55** |

**TABLE II**

| method       | CIFAR-10 | CIFAR-100 | Tiny ImageNet | ImageNet-1K |
|--------------|----------|-----------|--------------|-------------|
|              | 400ep    | 800ep     | 400ep        | 800ep       | 1600ep | 1600ep |
| Supervised   |          |           |              |             |        |        |
|              | 39.55    | 94.39     | 80.01        | 78.08       | 91.51  | 70.97  | 70.56 |
| Rotation [32] | 74.81    | 76.32     | 45.52        | 47.82       | 23.46  | 25.17  | 39.85  | 48.25 |
| NFID [38]    | 79.53    | 82.70     | 54.52        | 57.16       | 36.86  | 38.24  | 43.10  | 58.87 |
| ODC [35]     | 78.23    | 79.91     | 48.04        | 52.17       | 27.30  | 28.79  | 45.17  | 53.40 |
| SimCLR [40]  | 86.22    | 88.24     | 64.45        | 57.42       | 37.64  | 38.46  | 51.03  | 66.67 |
| MoCo.v2 [56] | 82.41    | 88.62     | 65.65        | 61.48       | 33.00  | 37.49  | 52.87  | 67.85 |
| BYOL [41]    | 82.61    | 88.15     | 62.32        | 64.40       | 33.93  | **38.81** | 54.62 | 71.88 |
| BarlowTwins [43] | 82.28 | 88.36     | 62.72        | 61.92       | 33.27  | 38.34  | 53.23  | 71.66 |
| GenURL       | **88.27** | **88.95** | **59.01**    | **63.51**   | **37.81** | **38.68** | **55.12** | **72.15** |
TABLE III
TRANSFER LEARNING ON CIFAR-10 CLASSIFICATION. TOP-1 ACCURACY (%) UNDER LINEAR EVALUATION IS REPORTED

| method          | 400 ep | 800 ep | 1600 ep |
|-----------------|--------|--------|---------|
| Related loc [30]| 66.41  | 69.34  | -       |
| Rotation [32]   | 71.01  | 64.29  | -       |
| NPID [38]       | 71.15  | 63.72  | 65.30   |
| ODC [35]        | 68.59  | 66.13  | 70.51   |
| SimCLR [40]     | 75.97  | 75.08  | 76.86   |
| MoCo.v2 [84]    | 74.46  | 76.54  | 75.61   |
| BYOL [41]       | 74.04  | 76.85  | 75.55   |
| SwAV [37]       | 74.17  | 76.28  | 76.34   |
| BiaNarTwins [43]| 74.63  | 76.71  | 76.12   |
| GenURL          | 80.22  | 80.12  | 79.85   |

TABLE IV
LOSS FUNCTION ANALYSIS ON SELF-SUPERVISED LEARNING. WE EVALUATE THE LOSS FUNCTIONS PROPOSED IN SECTION III ON STL-10, CIFAR-10/100, AND TINY IMAGENET. TOP-1 ACCURACY (%) UNDER LINEAR EVALUATION IS REPORTED.

| Loss function | p_X setting | STL-10 | CIFAR-10 | CIFAR-100 | Tiny |
|---------------|-------------|--------|----------|-----------|------|
| MSE           | static      | 88.72  | 84.26    | 57.31     | 36.82|
| BCE           | static      | 91.05  | 88.35    | 60.08     | 38.19|
| BCE           | dynamic     | 91.60  | 88.87    | 61.16     | 38.85|
| GKL           | static      | 91.21  | 88.63    | 61.27     | 38.07|
| GKL           | dynamic     | 91.85  | 89.95    | 61.51     | 38.48|

UMAP [4] and DMT [16] based on DMT implementation. Similar to the setting of GE tasks, we conduct a grid search of $v_Z, \mu_Z$, and $\sigma_Z$. We use $v_Z = 0.001$ and $\sigma_X = 5$ for MNIST and FMNIST datasets while using $v_Z = 0.01$ and $\sigma_X = 20$ for COIL-20.

B. Self-Supervised Visual Representation

In this section, we compare GenURL with three types of the existing self-supervised methods, including headcraft, clustering-based, and CL methods. For a fair comparison, we apply the same augmentation settings described in MoCo.v2 [84] to all CL methods and follow hyperparameters described in their original papers. We remove the Gaussian blur augmentation in CIFAR experiments [41], [42]. We perform unsupervised pretraining using ResNet-50 on STL-10 and ImageNet-1k, and using ResNet-18 on CIFAR-10/100, Tiny ImageNet, and ImageNet-1k. GenURL adopts the GKL loss and the dynamic setting in the SSL task.

1) Evaluation Protocols: As for linear evaluation, we follow the experiment settings in [39] to train a linear classifier for 100 epochs and use different base lr for different datasets. We set the base lr $= 1.0$ for STL-10 and CIFAR-100, lr $= 0.1$ for CIFAR-10 and Tiny, and lr $= 0.01$ for IN-1k. The learning rate decays by 0.1 at 60 and 80 epochs. As for semisupervised evaluation, we fine-tune the whole pretrained model for 20 epochs on STL-10 with a step schedule at 12 and 16 epochs, and batch size is 256. We perform grid search for each test methods on base lr $= \{0.1, 0.01, 0.001\}$ and parameterwise lr mut $= \{1, 10, 100\}$ of the fc layer. Both experiments use the SGD optimizer with the weight decay of 0 for linear evaluation and 0.0001 for semisupervised. Top-1 and top-5 accuracy are reported on the validation set.

2) Linear Evaluation Results: We first compare with existing methods in terms of different training epochs on STL-10, as shown in Table I. The proposed GenURL achieves the highest accuracy among all settings. It not only converges faster than other algorithms under 400-epoch pretraining but also gains better performance when training longer. Then, we compare various methods on CIFAR-10/100 and Tiny, as shown in Table II. GenURL achieves the top performance on three datasets under 400-epoch pretraining and achieves the second best on CIFAR-100 and Tiny ImageNet when training 800 epochs. Different from the existing contrastive-based methods, GenURL takes more pairwise relations between samples into consideration, which might help GenURL convergence faster. For example, given a mini-batch of N samples, BYOL utilizes $2N$ sample pairs, and MoCo requires $K + N$ sample pairs ($K$ is the memory bank size), while GenURL optimizes $N^2$ sample pairs (similar to SimCLR [40]).

3) Semisupervised Evaluation Results: In Table I, we fine-tune a ResNet-50 pretrained with various methods on the labeled training set of STL-10. GenURL outperforms other methods under 400-epoch and 800-epoch pretraining, which reflects its fast convergence speeds while maintaining the second-best classification accuracy with longer training.

4) Transferring Features: The main goal of unsupervised learning is to learn transferrable features. In Table III, we compare the representation quality of unsupervised pretraining on STL-10 by transferring to the classification task. We adopt linear evaluation on the CIFAR-10 in 64 × 64 resolutions with 1600-epoch pretrained ResNet-50 on STL-10, and other settings are the same as Section V-B. GenURL achieves the highest accuracy among all methods: +3.36% / +3.29% / +2.99% for GenURL pretraining 400/800/1600 epochs over the second-best method.

5) Ablation Studies for SSL Tasks: We first ablate the loss functions used in visual SSL tasks. Since the input distance in SSL tasks is only well defined for positive pairs where it can be optimized explicitly, the relationship between negative pairs can be implicitly modeled by the dynamic setting. As shown in Table IV, GenURL prefers the GKL loss when using the
static $\tilde{p}_X$, while the BCE loss yields better performance when using the dynamic to mine the relation between negative pairs. Then, we analyze hyperparameters of GenURL in Fig. 5. We find that GenURL prefers $\nu = 100$ and $\sigma = 1$ (approximating a standard Gaussian kernel) with a small batch size of 256. Moreover, we compare learned representations of GenURL with other visual SSL methods on STL-10 by UMAP [4] visualization in Fig. 6.

C. Unsupervised KD

We evaluate the KD tasks based on self-supervised learning on STL-10 dataset. In this experiment, we adopt MoCo.v2 with ResNet-50 under 1600-epoch pretraining. We choose multiple smaller networks with fewer parameters as the student network: ResNet-18 [70], MobileNet.v2 [86], and ShuffleNet.v1 [87]. Similar to the pretraining for the teacher network, we add one additional MLP layer on the basis of the student network. Following the linear evaluation protocols in Section V-B, we compare the existing relation-based KD methods, including RKD [65], PKT [64], SP [66], SSKD [68], CRD [69], and SEED [67]. We adopt the BCE loss for distillation than larger networks. From the perspective of student models, as shown in Table V, we notice that smaller methods, including RKD [65], PKT [64], SP [66], SSKD [68], CRD [69], and SEED [67]. We adopt the BCE loss for

![Fig. 7. Ablation of $\nu_z$, $\sigma$ and batch size of GenURL for KD tasks on STL-10.](image)

| KD methods | KD loss | ResNet-18 SSL | KD KD+SSL | MobileNet.v2 SSL | KD KD+SSL | ShuffleNet.v1 SSL | KD KD+SSL |
|------------|---------|---------------|-----------|------------------|-----------|------------------|-----------|
| RKD [65]   | H+AW   | 86.48         | 86.76     | 85.89            | 86.20     | 84.31            | 83.01     |
| PKT [64]   | KL     | 86.89         | 87.12     | 86.14            | 86.48     | 84.25            | 84.82     |
| SP [66]    | MSEL   | 86.53         | 86.74     | 85.96            | 86.13     | 84.22            | 84.76     |
| SSKD [68]  | KL+InfoNCE | 81.51       | 79.96     | 86.80            | 77.26     | 85.23            |           |
| CRD [69]   | KL+InfoNCE | -           | -         | -                | -         | -                | -         |
| SEED [67]  | KL+InfoNCE | -           | -         | -                | -         | -                | -         |
| GenURL     | BCE    | 88.05         | 88.13     | 86.61            | 86.85     | 84.67            | 85.10     |
| GenURL††   | BCE    | 88.26         | 88.39     | 87.28            | 87.47     | 85.05            | 85.38     |

D. Unsupervised GE

1) Setups and Results: Unsupervised GE experiments are conducted on three graph network datasets (CORA, CiteSeer, and PubMed), and we evaluate the learned embeddings by the node classification task. We compare GE methods that utilize both features and graph structures, including AGC [25], AGE [28], GIC [88], and ARGA [27]. The learned node embeddings are passed to logistic regression, and we report the mean and STD of linear classification accuracy (Acc) of comparison methods. Table VI shows that GenURL (BCE) using both graphs and attributed features achieves new state-of-the-art performances on three GE datasets and improves previous GE methods by at least 0.5%, 0.4%, and 1.0% top-1 accuracy on CORA, CiteSeer, and PubMed datasets.

2) Ablation and Analysis: We first adopt two ablation studies of the loss functions used in GenURL on GE tasks in Table VI: 1) when the input is both (using both graphs and attributed features with the dynamic $\tilde{p}_X$), the BCE loss shows better performance than the GKL loss and 2) when using the BCE loss, using both with the dynamic $\tilde{p}_X$ outperforms feature (only using the attributed features) with the static $\tilde{p}_X$. Then, we visualize the learned embedding on CiteSeer by UMAP in Fig. 8. We find that GenURL separates subgraphs...
We visualize the last latent space of the encoder by UMAP. The result of GenURL contains both the topology and the local geometric structures.

Table VII shows the DR, Trust, Cont, and Top-1 Accuracy (%) are reported on MNIST, FMNIST, and COIL-20 datasets.

E. Dimension Reduction

1) Setups and Results: We perform DR experiments on MNIST, FMNIST, and COIL-20 datasets. We compare the current leading methods, including nonparametric methods (t-SNE [13] and UMAP [4]) and parametric methods (P-UMAP [15], GRAE [89], TopoAE [11], and DMT [16]). Besides the linear classification top-1 accuracy (Acc) with logistic regression, we evaluate the qualities of the low-dimensional representation in terms of the input space with Trust and Cont. Since the DR task only relies on the input space, GenURL adopts the BCE loss and the static $\tilde{p}_X$.

As shown in Table VII, we compare GenURL (BCE) with the existing DR methods and find that GenURL yields comparable performance in terms of Trust and Acc, which indicates that GenURL (BCE) keeps the balance between local geometric structures (achieving better Trust and Cont) and the distinction of different submanifolds (achieving better linear classification accuracy).

2) Ablation and Analysis: We first conduct the ablation of the BCE or GKL losses in Table VII: GenURL (BCE) always outperforms GenURL (GKL) on three DR datasets, because we adopt a large batch size as DMT and P-UMAP. Then, we provide DR results on COIL-20 in Fig. 10 and find that GenURL captures more geometric structures than previous methods, especially UMAP and TopoAE (focusing on topological structures). Moreover, we ablate hyperparameters of GenURL (BCE) on MNIST in Fig. 11 and find that GenURL prefers $\sigma = 1$, $\nu_Z = 0.001$, and the batch size of 2048.

F. Analysis and Discussion

We provide an empirical analysis of the hyperparameters and loss functions in GenURL on different URL tasks to demonstrate the characteristics of various tasks. We compare the results using different batch sizes, $\nu_Z$, $\sigma$, and loss functions used in GenURL to demonstrate the relationship of SSL, KD, and DR tasks.

1) Relationship Between DR and GE: As shown in Figs. 9 and 12, GenURL prefers smaller $\nu_Z$ for both the GE and DR tasks, because the large $\nu_Z$ yields crowd embedding while the small $\nu_Z$ conducts separable results. Therefore, we consider DR as a special type of GE task. The main difference between DR and GE tasks is GE that takes both the node and edge features into consideration.

2) Relationship Between SSL and KD: Then, we compare how GenURL deals with the negative samples in SSL and KD tasks. In Fig. 7, we find that GenURL prefers the similar...
v_Z and σ for both SSL and KD tasks, which indicates using v_Z = 100 and σ = 1 (the standard Gaussian kernel) is suitable for L2-normalized cosine distance. Notice that GenURL prefers small batch sizes, such as 256, for the SSL task (suffering performance drops when the batch size increases) while prefers larger batch sizes for the KD task. It might be because pairwise similarities of negative samples in SSL tasks are unreliable and can be regarded as dark knowledge in the KD task [59], [67]. The gradient from negative pairs might overwhelm positive samples at the early training stage of the SSL task, while negative samples are well defined by the teacher model in the KD task. Meanwhile, Table VIII shows that using the dynamic version and the GKL loss in SSL tasks yield the best performance while using the large batch size, and the BCE loss in KD tasks performs better. Therefore, we conclude that the GKL with the dynamic structural modeling can alleviate the harmful effects of unreliable metric spaces of SSL tasks.

3) Relationship Between SSL and DR: We further discuss the relation between SSL and DR tasks (the dynamic and static ˜p_X) with GenURL to explain the desired structures of data in URL tasks. Meanwhile, we find that the instance discrimination prior knowledge in SSL tasks is more useful for downstream tasks, such as clustering and classification of complex scenarios, as shown in Fig. 13. Therefore, we can choose a proper URL task for various scenarios: the DR task is suitable for MNIST and FMNIST datasets where the input spaces are discriminative and reliable, while the SSL task suits more complex datasets, such as CIFAR-10/100, where the input spaces are unreliable.

4) Hyperparameters: First, we compare the effects of hyperparameters in GenURL for DR and SSL tasks. As shown in Fig. 12, GenURL prefers smaller v_Z, i.e., using v = 0.01 to balance the local and global structures. Fig. 5 shows that GenURL prefers v_Z = 100 for representations with strong discriminative abilities. Then, we find that the prior knowledge of instance discrimination in SSL tasks is more useful for downstream tasks, such as clustering and classification of complex scenarios. As shown in Fig. 13, we summarize the learned representations with DR and SSL methods on CIFAR-10 and find that the results of SSL (adopting the instance discrimination prior knowledge) are more reliable and useful to downstream tasks, such as clustering and classification than DR. Empirically, DR methods are employed on “simple” datasets (e.g., MNIST and COIL-20) with reliable geometric structures rather than “complex” datasets (e.g., CIFAR-10 and ImageNet) with high-dimensional and redundant features. We hypothesize that this might depend on the property of the dataset. To verify our assumption, we compute the pairwise distance between raw input samples and the latent space of the SSL model, as shown in Fig. 14, to show the difference between DR and SSL tasks. We find that the input space is discriminative and reliable enough on MNIST and FMNIST for the DR task, while it is unreliable on CIFAR-10. Therefore, we can conclude as follows: when the dataset is reliable, GenURL can employ the BCE loss and the static ˜p_X to perform DR tasks (or GE tasks on the graph data); when the dataset contains high-dimensional redundant features, GenURL can adopt the GKL loss and the dynamic ˜p_X with the prior knowledge to conduct SSL tasks.

5) Complexities: GenURL has a constant algorithmic complexity in the four URL tasks because of its property of unification. It improves performance without adding extra complexity. Unlike GNN and GCN, the proposed GenURL does not require neighborhood aggregation operations, making the complexity agnostic to the network architectures and the kNN graph in the input space. In Section III, we build an undirected graph before training and calculate the adjacency matrix A with O(n^2). The precomputed results will be saved to get p_X(x_i, x_j) in (3). In each iteration, we calculate the latent space pairwise similarity p_X(z_i, z_j) with O(n^2), assuming the batch size is n (i.e., the whole dataset in GE and DR tasks). In CL and KD tasks, we will practically use a mini-batch of b to reduce the computational complexity to O(b^2).

VI. LIMITATION AND FUTURE WORK
As for the societal impacts of GenURL, it can be regarded as a unified framework for the URL problem that bridges the gap between various methods. The ablation studies of basic hyperparameters can reflect the relationship between different URL tasks. The core idea of GenURL is to explore intrinsic structures of the data (the raw input space or empirical metric space) and preserve these structures in the latent space, which might inspire some improvements in various URL tasks. For example, the dynamic ˜p_X is similar to the hard negative mining problem in the SSL task [47], [74].

As for the limitations of GenURL, we can conclude three aspects: 1) the proposed framework relies on offline
hyperparameter tuning to adapt to new URL tasks, which makes it tough to handle more than two input similarities; 2) GenURL cannot deal with the case of discrete empirical spaces well, e.g., the SSL tasks, and the dynamic $\hat{p}_X$ should be improved in the future work; and 3) the performance of GenURL is still limited by negative samples (sensitive to the size of datasets and the batch size). In future work, we plan to tackle the aforementioned limitations and design a dynamic framework that can learn to optimize the DSM and low-dimensional embedding in an end-to-end manner. Moreover, GenURL will be used in more URL application scenarios.

VII. CONCLUSION

We propose a simple but efficient similarity-based framework for URL, called GenURL, that encodes essential structures from the input data and optional prior knowledge. Specifically, we discuss the loss functions for embedding learning in GenURL and proposed BCE loss and GKL divergence loss. Combined with a specific pretext task, we can adapt GenURL to various URL scenarios in a unified manner and achieve state-of-the-art performances, including self-supervised visual representation learning, unsupervised KD, GEs, and DR. Moreover, ablation studies reflect the relationship between data characters and hyperparameter settings of GenURL.

ACKNOWLEDGMENT

This work was done when Zhiyuan Chen interned at Westlake University, Hangzhou, China.

REFERENCES

[1] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (coil-20),” Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-005-06, 1996. [Online]. Available: https://www.cs.columbia.edu/CVlEsoftware/software/coil-20.php
[2] C. L. Giles, K. D. Bollacker, and S. Lawrence, “CiteSeer: An automatic citation indexing system,” in Proc. 3rd ACM Conf. Digit. Libraries, 1998, pp. 89–98, doi: 10.1145/276675.276685
[3] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in Proc. 14th Int. Conf. Artif. Intell. Statist., 2011, pp. 215–223.
[4] L. McMinn, J. Healy, and J. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” 2018, arXiv:1802.03436.
[5] N. Mu, A. Kirillov, D. Wagner, and S. Xie, “SLIP: Self-supervision meets language-image pre-training,” 2021, arXiv:2112.12750.
[6] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “Data2Vec: A general framework for self-supervised learning in speech, vision and language,” 2022, arXiv:2205.03555.
[7] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
[8] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[9] Z. Zhang and H. Zha, “Principal manifolds and nonlinear dimensionality reduction via tangent space alignment,” SIAM J. Sci. Comput., vol. 26, no. 1, pp. 313–338, Jan. 2004.
[10] D. L. Donoho and C. Grimes, “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data,” Proc. Nat. Acad. Sci. USA, vol. 100, no. 10, pp. 5591–5596, May 2003.
[11] M. Moor, M. Horn, B. Rieck, and K. Borgwardt, “Topological autoencoders,” in Proc. Int. Conf. Mach. Learn. (ICML), 2020.
[12] S. Melacci and M. Gori, “Unsupervised learning by minimal entropy encoding,” IEEE Trans. Neural Netw. Learn. Syst., vol. 23, no. 12, pp. 1849–1861, Dec. 2012.
[13] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 11, pp. 2579–2605, 2008.
[14] L. van der Maaten, “Learning a parametric embedding by preserving local structure,” J. Mach. Learn. Res., vol. 5, pp. 384–391, Jan. 2009.
[15] T. Sainburg, L. McInnes, and T. Q. Gentner, “Parametric UMAP embeddings for representation and. semi-supervised learning,” Neural Comput. 33, no., vol. 11, no. 2021, pp. 2881–2907., 2021.
[16] S. Z. Li, Z. Zang, and L. Wu, “Deep manifold transformation for nonlinear dimensionality reduction,” 2020, arXiv:2010.14831.
[17] S. Li, H. Lin, Z. Zang, L. Wu, J. Xia, and S. Z. Li, “Invertible manifold learning for dimension reduction,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML-PKDD), 2021, pp. 713–728.
[18] M. E. J. Newman, “Finding community structure in networks using the eigenvectors of matrices,” 2006, arXiv:physics/0605087.
[19] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in Proc. Int. Conf. Mach. Learn. (ICML), 2014.
[20] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2014.
[21] R. Zhang, Y. Zhang, and X. Li, “Unsupervised feature selection via adaptive graph learning and constraint,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 3, pp. 1355–1362, Mar. 2022.
[22] Y. Li and C. Sha, “Community detection in attributed graphs: An embedding approach,” in Proc. AAAI Conf. Artif. Intell. (AAAI), 2018, pp. 338–345.
[23] A. Bojchevska and S. G"unnewann, “Bayesian robust attributed graph clustering: Joint learning of partial anomalies and group structure,” in Proc. AAAI Conf. Artif. Intell. (AAAI), 2018, pp. 265–271.
[24] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” 2016, arXiv:1611.07308.
[25] X. Zhang, H. Liu, Q. Li, and X.-M. Wu, “Attributed graph clustering via adaptive graph convolution,” 2019, arXiv:1906.01210.
[26] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, arXiv:1609.02907.
[27] S. Pan, R. Hu, S.-F. Fung, G. Long, J. Jiang, and C. Zhang, “Learning graph embedding with adversarial training methods,” IEEE Trans. Cybern., vol. 50, no. 6, pp. 2475–2487, Jun. 2020.
[28] G. Cui, J. Zhou, C. Yang, and Z. Liu, “Adaptive graph encoder for attributed graph embedding,” in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2020.
[29] Z. Zang, S. Li, D. Wu, J. Guo, Y. Xu, and S. Z. Li, “Unsupervised deep manifold attributed graph embedding,” 2021, arXiv:2104.13048.
[30] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1422–1430.
[31] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018.
[32] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2018.
[33] D. He et al., “Analyzing heterogeneous networks with missing attributes by unsupervised contrastive learning,” IEEE Trans. Neural Netw. Learn. Syst., pp. 1–13, 2022.
[34] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018.
[35] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, “Online deep clustering for unsupervised representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6687–6696.
[36] Y. M. Asano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2020.
[37] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrastive cluster assignments,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.
[38] Z. Wu, Y. Xiong, S. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance-level discrimination,” 2018, arXiv:1805.01978.
[39] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9726–9735.
[40] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020, arXiv:2002.05709.
J.-B. Grill et al., “Bootstrap your own latent: A new approach to self-supervised learning,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

X. Chen and K. He, “Exploring simple Siamese representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 15747–15753.

J. Zhontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” in Proc. Int. Conf. Mach. Learn. (ICML), 2021, pp. 12310–12320.

Z. Zang et al., “DLMoE: Deep local-flatness manifold embedding,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2022.

C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, “Z. Zang et al., “DLME: Deep local-flatness manifold embedding,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020, pp. 8765–8775.

Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, “Hard negative mixing for contrastive learning,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

J. D. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, “Contrastive learning with hard negative samples,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.

S. Li, Z. Liu, Z. Wang, D. Wu, Z. Liu, and S. Z. Li, “Boosting discriminative visual representation learning with scenario-agnostic mixup,” 2021, arXiv:2111.15454.

E. Xie et al., “DetCo: Unsupervised contrastive learning for object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 8372–8381.

T. Xiao, C. J. Reed, X. Wang, K. Keutzer, and T. Darrell, “Region similarity representation learning,” 2021, arXiv:2103.12902.

R. R. Selvaraju, K. Desai, J. Johnson, and N. Nair, “CASTing your model: Learning to localize improves self-supervised representations,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11053–11062.

D. Wu, S. Li, Z. Zang, and S. Z. Li, “Exploring localization for self-supervised fine-grained contrastive learning,” in Proc. Brit. Mach. Vis. Conf. (BMVC), 2022.

A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.

Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 9993–10002.

H. Bao, L. Dong, and F. Wei, “BET: BERT pre-training of image transformers,” in Proc. Int. Conf. Learn. Representations (ICLR), 2022.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” 2021, arXiv:2111.06377.

Z. Xie et al., “SimMIM: A simple framework for masked image modeling,” 2021, arXiv:2111.09886.

S. Li, D. Wu, F. Wu, Z. Zang, and S. Z. Li, “Architecture-agnostic image modeling - from ViT back to CNN,” 2022, arXiv:2205.13943.

G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, arXiv:1503.02531.

Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning,” 2017, arXiv:1706.03884.

Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves ImageNet classification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10684–10695.

S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017.

J. Yin, D. Joo, J. Bae, and J. Kim, “A gift from knowledge distillation: Faster optimization, network minimization and transfer learning,” in Proc. IEEE. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 7130–7138.

N. Passalis and A. Tefas, “Learning deep representations with probabilistic knowledge transfer,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018.

J. Park, D. Kim, Y. Lu, and M. Cho, “Relational knowledge distillation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3962–3971.

F. Tung and G. Mori, “Similarity-preserving knowledge distillation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1365–1374.

Z. Fang, J. Wang, L. Wang, L. Zhang, Y. Yang, and Z. Liu, “SEED: Self-supervised distillation for visual representation,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.
Zicheng Liu (Student Member, IEEE) received the B.S. degree from the Department of Information and Computing Science, University of Liverpool, Liverpool, U.K., in 2020. He is currently pursuing the Ph.D. degree with the School of Engineering, Westlake University, Hangzhou, China, and Zhejiang University, Hangzhou, under the supervision of Prof. Stan Z. Li.

His research interests include data augmentation, network architecture design, and biological and computer vision applications.

Zelin Zang (Graduate Student Member, IEEE) received the M.Eng. degree from the Zhejiang University of Technology, Hangzhou, China, in 2020. He is currently pursuing the Ph.D. degree with Westlake University, Hangzhou, China, and Zhejiang University, Hangzhou, under the supervision of Prof. Stan Z. Li.

His research interests include manifold learning, dimension reduction, self-supervised learning in computer vision, and biological applications.

Di Wu received the B.S. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in 2016, and the M.S. degree in electrical and computer engineering from Boston University, Boston, MA, USA, in 2018. He is currently pursuing the Ph.D. degree with the Center of Excellence in Biomedical Research on Advanced Integrated-on-Chips Neurotechnologies, School of Engineering, Westlake University, Hangzhou, China.

His research interests include self-supervised learning and efficient deep learning for neurophysiological applications.

Zhiyuan Chen received the B.S. degree from Australian National University, Canberra, ACT, Australia, in 2022. We worked as a Research Assistant of the StarBridge Program and a Research Intern of the Star of Tomorrow Program under the guidance of Dr. Pan Deng at Microsoft Research Asia, Beijing, China, and Microsoft Research AI for Science, Beijing, from 2021 to 2022, and also worked as a Research Assistant at Westlake University in 2020. He is currently a Researcher at Deep Potential, Beijing, China, working on macromolecules. His research interests include AI4Science, computer vision, and deep learning.

Stan Z. Li (Fellow, IEEE) received the B.Eng. degree from Hunan University, Changsha, China, in 1982, the M.Eng. degree from the National University of Defense Technology, Changsha, in 1985, and the Ph.D. degree from the University of Surrey, Guildford, U.K., in 1991.

He was the Director of the Center for Biometrics and Security Research (CBSR), Chinese Academy of Sciences, Beijing, China, from 2004 to 2019. He worked at Microsoft Research Asia, Beijing, as a Research Lead, from 2000 to 2004. Prior to that, he was an Associate Professor (Tenure) at Nanyang Technological University, Singapore. He joined Westlake University, Hangzhou, China, as a Chair Professor of artificial intelligence, in February 2019. He has published over 400 articles in international journals and conferences and authored and edited ten books, with over 50,000 Google Scholar citations. Among these are Markov Random Field Models in Image Analysis (Springer), Handbook of Face Recognition (Springer), and Encyclopedia of Biometrics (Springer).

His research interests include fundamental research in machine learning, data science, and applied research in multiple AI-related interdisciplinary fields (computer vision, smart sensors, life science, material science, and environmental science).

Dr. Li served as an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and organized more than 100 international conferences or workshops.