Probabilistic Structural Latent Representation for Unsupervised Embedding

Mang Ye, Jianbing Shen*
Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
{mangye16, shenjianbingcg}@gmail.com

Abstract

Unsupervised embedding learning aims at extracting low-dimensional visually meaningful representations from large-scale unlabeled images, which can then be directly used for similarity-based search. This task faces two major challenges: 1) mining positive supervision from highly similar fine-grained classes and 2) generating to unseen testing categories. To tackle these issues, this paper proposes a probabilistic structural latent representation (PSLR), which incorporates an adaptable softmax embedding to approximate the positive concentrated and negative instance separated properties in the graph latent space. It improves the discriminability by enlarging the positive/negative difference without introducing any additional computational cost while maintaining high learning efficiency. To address the limited supervision using data augmentation, a smooth variational reconstruction loss is introduced by modeling the intra-instance variance, which improves the robustness. Extensive experiments demonstrate the superiority of PSLR over state-of-the-art unsupervised methods on both seen and unseen categories with cosine similarity. Code is available at https://github.com/mangye16/PSLR

1. Introduction

Supervised embedding learning focuses on optimizing a network in which the low-dimensional features belonging to the same class are concentrated, while features from different classes are separated [33, 35, 48, 61, 29]. Powerful supervised learning models have achieved human-level performance in various tasks, such as face recognition [32] and person re-identification [55]. However, enough annotated data needed for supervised methods requires extensive human efforts. Consequently, this paper addresses the unsupervised embedding learning (UEL) problem [56], learning discriminative representations without human annotation.

UEL requires that the similarity between learned features is consistent with the visual similarity/category relations of input images, which can be subsequently used for similarity-based search (as shown in Fig. 1). In comparison, the general unsupervised feature learning (UFL) [4, 7, 30, 34, 37, 50, 57] mainly focuses on learning good "intermediate" features for downstream tasks, *e.g.* train linear classifiers or object detectors using the unsupervisedly learned features from a subset of labeled images. However, the learned features may not preserve visual similarity, *i.e.* the performance drops dramatically for similarity search [56].

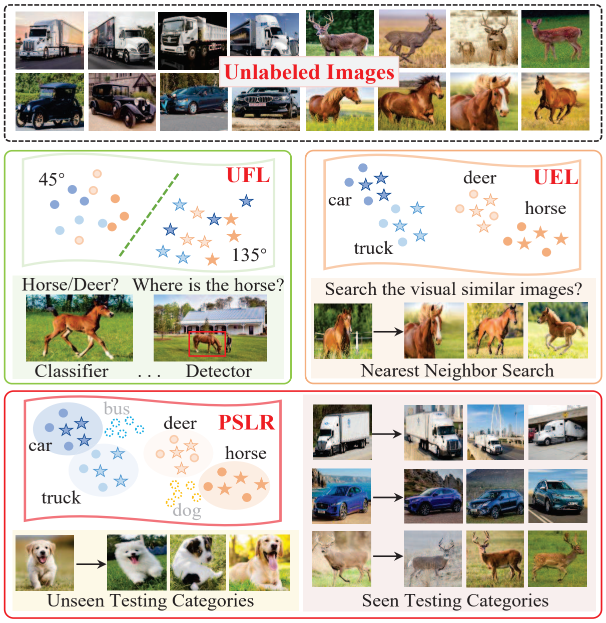


Figure 1: Comparison between the general UFL, UEL and the proposed PSLR. UFL usually focuses on learning linear separable "intermediate" features using supervision signal, *e.g.*, rotation in [12, 37]. The learned features may not preserve visual consistency, while UEL aims at extracting visually meaningful representations for similarity-based search. In contrast, our PSLR optimizes the latent representation with intra-instance variation modeling to enhance the generalizability on unseen testing categories.

The major challenge in UEL is to mine the visual similarity relationship or weak positive supervision from unlabeled images. Following supervised embedding learning, MOM [20] was developed to mine hard positive and negative samples in the manifold space. However, its la-

[*] Corresponding author: *Jianbing Shen*.

bel mining relies heavily on the initialized representation. Instance-wise supervision is another popular approach for UEL [19, 51, 56]. Specifically, different instances are treated as negative samples and purposely separated in the embedding space [3, 51]. Along a similar line, anchor neighborhood discovery (AND) [19] was proposed to enhance the positive similarity with the mined nearest neighbors [38]. However, the neighborhood discovery may introduce a large number of false positives, especially in fine-grained image recognition tasks (§4.2). Another drawback is that their optimization is performed on prototype memory [19, 51] rather than the instance features, which results in limited efficiency. Similarly, an augmentation invariant and spreading instance feature (ISIF) was introduced in [56], where random data augmentation was applied to provide positive supervision. However, data augmentation can only provide limited positive supervision, and over-fitting to these augmented instance features will result in poor generalizability, *i.e*., the learned representation does not perform well when training and testing categories do not overlap (*unseen testing categories*) with unknown variations.

This paper presents a novel probabilistic structural latent representation (PSLR) for UEL. Specifically, PSLR mines the relationship within each training batch by learning a graph latent representation with variational structural modeling, which approximates the data augmentation concentrated and negative instance separated properties in the latent space. A novel adaptable softmax embedding is introduced to optimize the latent representations rather than the instance features. This results in better generalizability on unseen testing categories while maintaining high learning efficiency. By enlarging the discrepancy between the positive and negative sample pairs using an adaptable factor, the discriminability is reinforced without introducing additional computational cost. It also significantly improves the performance of the ISIF method [56]. Moreover, PSLR incorporates with a smooth variational self-reconstruction loss to enhance the robustness against image noise. This strategy also improves the generalizability on unseen testing categories by applying auxiliary noise to the latent representation, which enriches the positive supervision.

Our main contributions are summarized as follows: We propose a novel probabilistic structural latent representation (PSLR) for unsupervised embedding learning. The optimization on the latent representation results in higher accuracy than competing methods, while it maintains high learning efficiency compared to the direct representation optimization. We introduce an adaptable softmax embedding on latent representation by enlarging the positive/negative difference. This provides stronger discriminability and better generalizability without additional cost. We outperform the current state-of-the-art on five datasets under both seen and unseen testing categories with cosine similarity search.

2. Related Work

Unsupervised Deep Learning. There are four main approaches for unsupervised deep learning [4], as follows: 1) *Estimating Between-image Labels*, this approach mines the between-image relationship with clustering [4, 10, 30] or nearest neighbors [19, 44] to provide label information. 2) *Generative Model*, it usually learns the true data distribution with a parameterized mapping. The most commonly used models include Bolztmann Machines (RBMs) [27, 43], Auto-encoders [18, 45, 57] and generative adversarial network (GAN) [13, 8, 11]. 3) *Self-supervised Learning,* this approach designs supervision signals to guide feature learning [21, 24], such as the context information of local patches [7], the position of randomly rearranged patches [34, 50], the missing pixels of an image [36], the color patterns [58] and spatial-temporal information in videos [1, 47]. 4) *Instance-wise Learning,* it treats each image instance as a distinct class by separating the different instance features [9, 51, 56] or local aggregation [19, 62].

Most of the above methods belong to general unsupervised feature learning, where the learned representation is applied to downstream tasks with a small set of annotated training samples. However, the learned representation may not preserve visual meaning [56], making them unsuitable for similarity-based tasks, *i.e*., *nearest neighbor search*, *person re-identification* [52, 53, 54].

Unsupervised Embedding Learning. This approach aims at learning a visually meaningful representation by optimizing the similarity between samples. With a proper initialized representation, Iscen *et al.* [20] mined hard positive and negative samples in the manifold space and then the embedding is trained with triplet loss. Later, an augmentation invariant and spreading instance feature (ISIF) [56] was introduced for UEL. The challenging unseen testing categories require additional generalizability rather than overfitting to the seen training categories.

Our method is closely related to the graph variational auto-encoder [23, 60], utilizing the structural relationships among input graph nodes. It is also related to variational deep metric learning [31, 39]. However, our method is entirely unsupervised, without any input edge information.

3. The Proposed PSLR Method

Problem Formulation. Given a set of n unlabeled images $X = \{x_1, x_2, \cdots, x_n\}$, UEL aims at learning a feature extraction network $f_\theta(\cdot)$, which maps the input image x_i into a low-dimensional embedding feature $f_\theta(x_i) \in \mathbb{R}^{1\times d}$ (d is the feature dimension). For simplicity of notation, the instance feature representation $f_\theta(x_i)$ of an input image x_i is represented by $\mathbf{x}_i \in \mathbb{R}^{1\times d}$. As pointed out in [35, 41], the learned embedding should satisfy two properties: *positive concentration* and *negative separation*.

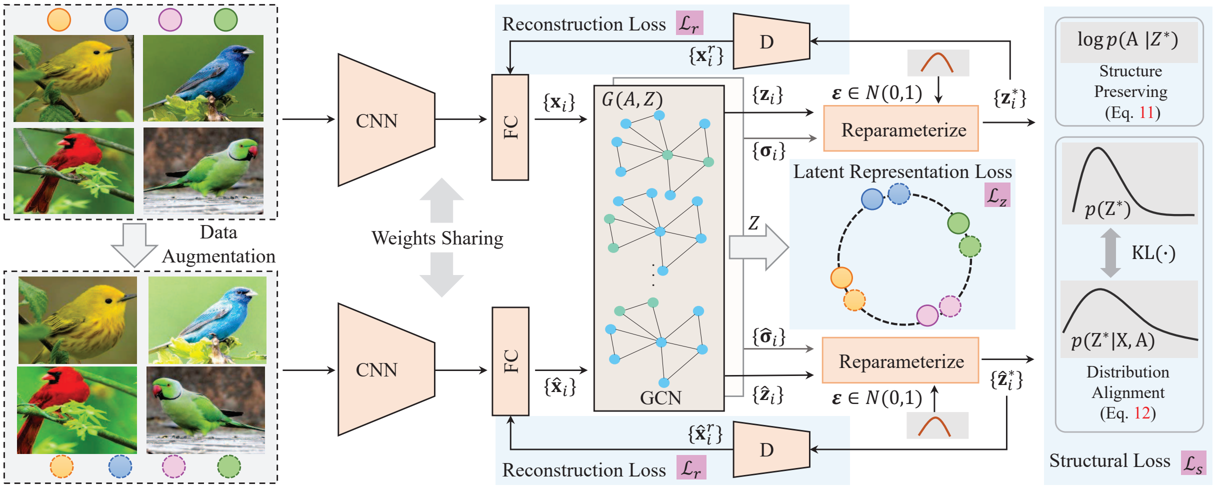

Data
Augmentation
CNN
FC
CNN
FC
Weights Sharing
Reconstruction Loss $\mathcal{L}_r$
D
$\{\mathbf{x}_i^r\}$
$\{\mathbf{x}_i\}$
$G(A,Z)$
GCN
$\{\mathbf{z}_i\}$
$\{\boldsymbol{\sigma}_i\}$
$\boldsymbol{\varepsilon} \in N(0,1)$
Reparameterize
$\{\mathbf{z}_i^*\}$
Latent Representation Loss $\mathcal{L}_z$
Z
$\{\hat{\boldsymbol{\sigma}}_i\}$
$\{\hat{\mathbf{z}}_i\}$
$\boldsymbol{\varepsilon} \in N(0,1)$
Reparameterize
$\{\hat{\mathbf{z}}_i^*\}$
$\{\hat{\mathbf{x}}_i\}$
$\{\hat{\mathbf{x}}_i^r\}$
D
Reconstruction Loss $\mathcal{L}_r$
$\log p(\mathrm{A}\,|Z^*)$
Structure
Preserving
(Eq. 11)
$p(Z^*)$
KL(·)
$p(Z^*|\mathrm{X},\mathrm{A})$
Distribution
Alignment
(Eq. 12)
Structural Loss $\mathcal{L}_s$

Figure 2: Overview of PSLR trained with a Siamese network. The feature embedding network projects the input images into low-dimensional normalized features. PSLR approximates the data augmentation invariant and instance separating properties with adaptable softmax embedding on latent space in §3.2, together with self-reconstruction in §3.3 and the probabilistic structural preserving in §3.4 .

Without class-wise labels, we approximate the above two properties using the data augmentation as positive supervision, *i.e.*, the features of the same instance under different data augmentations should be invariant, whilst features of different instances should be spread-out. Along this line, the proposed PSLR achieves better robustness against noisy instances and better generalizability to unseen testing categories. An overview of PSLR is shown in Fig. 2.

3.1. Graph Latent Representation

Our model takes the embedding instance features $\{\mathbf{x}_i\}$ as input, and the graph latent representation $\{\mathbf{z}_i\}$ is obtained using a graph convolutional network (GCN) by constructing an undirected graph $\mathcal{G}$ within each training batch.

At each training step, m instances $\{x_i\}_{i=1}^m$ are randomly sampled and data augmentation is performed to generate the augmented sample set $\{\hat{x}_i\}_{i=1}^m$. We represent the feature set of both the original and augmented features as $\{X = \mathbf{x}_1, \cdots, \mathbf{x}_m, \hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_m\} \in \mathbb{R}^{2m \times d}$. We construct an undirected graph $\mathcal{G} = (A, Z)$ using the relationship between the instance features within X, and the adjacency matrix $A \in \mathbb{R}^{2m \times 2m}$ is computed by

$$A = \mathbb{I}_{2m}, \tag{1}$$

where $\mathbb{I}_{2m}$ is an identity matrix, indicating that each node is connected to itself. The main reason is that it is difficult to mine the reliable structure relations without label information for graph construction. Note that neighborhood discovery (AND) [19] might also be adopted to enhance the graph construction using the mined additional positive information with neighbors (*e.g.*, on the CIFAR-10 dataset, as shown in § 4.1.1). However, this strategy suffers under fine-grained image recognition settings, since it is difficult to mine reliable positives. The graph latent representation Z is then obtained by a graph convolutional layer

$$Z = \phi(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X W), \tag{2}$$

where $D_{ii} = \sum_j A_{ij}$ is the degree matrix of A and $\phi(\cdot)$ represents the ReLU activation function. W is the network weight matrix. The graph latent representation $\{Z = \mathbf{z}_1, \cdots, \mathbf{z}_m, \ \hat{\mathbf{z}}_1, \cdots, \hat{\mathbf{z}}_m\} \in \mathbb{R}^{2m \times d}$ incorporates contextual information from the instance features. We can alternatively use a linear layer to obtain the latent representation.

3.2. Adaptable Softmax on Latent Representation

With the above graph latent representation, we propose a new adaptable softmax embedding method to approximate the *positive concentration* and *negative separation* properties. For each instance x_i, we treat the augmented latent representation $\hat{\mathbf{z}}_i$ as the positive sample, while the latent representations $\mathbf{z}_{k(k \neq i)}$ from other instances are considered as negatives. The probability of augmented sample $\hat{x}_i$ being recognized as instance x_i is represented by

$$P(i|\hat{x}_i) = \frac{\exp(\mathbf{z}_i^T \hat{\mathbf{z}}_i / \tau)}{\exp(\mathbf{z}_i^T \hat{\mathbf{z}}_i / \tau) + \eta \cdot \sum_{k \neq i} \exp(\mathbf{z}_k^T \hat{\mathbf{z}}_i / \tau)}, \tag{3}$$

where $\eta > 1$ is a magnification parameter to enlarge the similarity difference, enlarging the negative similarity contribution in the denominator. $\tau < 1$ is a temperature parameter to smooth the probability distribution [16, 19, 51]. Note that all the latent representations are ℓ_2 normalized for numerical stability, *i.e.*, $|\mathbf{z}_i|_2 = 1$.

Similarly, the probability of augmented sample $\hat{x}_i$ being recognized as instance $x_{j(j \neq i)}$ is calculated by

$$P(j|\hat{x}_i) = \frac{\exp(\mathbf{z}_j^T \hat{\mathbf{z}}_i / \tau)}{\exp(\mathbf{z}_i^T \hat{\mathbf{z}}_i / \tau) + \eta \cdot \sum_{k \neq i} \exp(\mathbf{z}_k^T \hat{\mathbf{z}}_i / \tau)}. \tag{4}$$

Finally, our adaptable softmax embedding on latent representation is formulated by minimizing the sum of the negative log likelihood over all instances, which is represented by

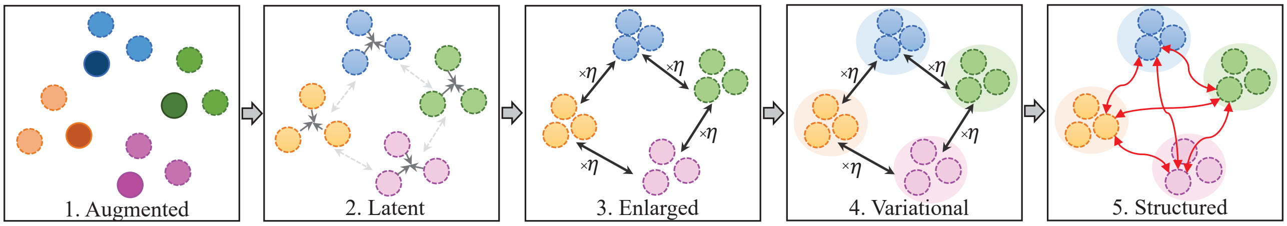


Figure 3: Step-by-step illustration of PSLR. Given the *augmented* instance features, we optimize the *latent* representations using the *enlarged* positive/negative similarity. *Variational* modeling and *structural* information are incorporated to reinforce the embedding learning.

$$\mathcal{L}_z = -\eta \cdot \sum_i \sum_{j \neq i} \log(1 - P(j|\hat{x}_i)) - \sum_i \log P(i|\hat{x}_i). \tag{5}$$

Our adaptable softmax embedding has two major advantages: 1) The adaptable factor $\eta > 1$ enlarges the discrepancy between the positive and negative similarities, which enhances the model's discriminability by addressing the imbalance between positive and negative sample pairs; 2) Performing softmax on the latent representation provides better generalizability to unseen testing categories, as demonstrated in § 4.2, since this modification prevents the network over-fitting the training instance features. In summary, the adaptable softmax embedding improves the accuracy while maintaining high efficiency by directly optimizing the latent representations, as illustrated in Fig. 4 in § 4.1.

3.3. Smooth Variational Self-Reconstruction

To enhance the robustness, we design a smooth variational self-reconstruction loss. The basic idea is to reconstruct the original input embedding features X using the noise corrupted latent representation (both original and augmented) $Z^* = \{\mathbf{z}_i^*\}_{i=1}^{2m}$, through a reparameterization process [22]. Specifically, we assume $\mathbf{z}_i^*$ satisfies the univariate Gaussian distribution, $\mathbf{z}_i^* \sim p(\mathbf{z}_i^*|x_i) = \mathcal{N}(\mathbf{z}_i, \boldsymbol{\sigma}_i^2)$. The reparametrized latent representation is then represented by

$$\mathbf{z}_i^* = \mathbf{z}_i + \boldsymbol{\sigma}_i \cdot \epsilon, \tag{6}$$

where $\boldsymbol{\sigma}_i$ is the output of another GCN layer based on $\mathbf{x}_i$. $\epsilon \sim \mathcal{N}(0, 1)$ is an auxiliary noise variable. To enhance the representational capacity of the embedding features, we add another decoder $D(\cdot)$ to reconstruct $\mathbf{x}_i$ based on $\mathbf{z}_i^*$, *i.e.* $\mathbf{x}_i^r = D(\mathbf{z}_i^*)$. Here, a smooth L1 loss is adopted as the reconstruction loss

$$\mathcal{L}_r = \sum_{i \in \mathcal{B}} \begin{cases} 0.5(\mathbf{x}_i - \mathbf{x}_i^r)^2, & |\mathbf{x}_i - \mathbf{x}_i^r| < 1 \\ |\mathbf{x}_i - \mathbf{x}_i^r|, & otherwise. \end{cases} \tag{7}$$

The variational self-reconstruction has two major advantages: it enhances the robustness by capturing the informative components [17, 49], and it simultaneously improves the discriminability by enriching the positive supervision besides the data augmentation. In addition, the smooth L1 loss is easy to optimize, ensuring a stable training.

3.4. Probabilistic Structural Preserving

This section presents a probabilistic structure preserving strategy to enhance the unsupervised embedding feature learning [23]. The structural loss $\mathcal{L}_s$ contains two main components: the *structure preserving* loss $\mathcal{L}_g$ and the *distribution alignment* loss $\mathcal{L}_{kl}$.

$$\mathcal{L}_s = \mathcal{L}_g + \mathcal{L}_{kl}. \tag{8}$$

Structure Preserving. This component matches the graph structure of Z^* (from both original and augmented samples) with the original graph input $\mathcal{G}$. Specifically, the structure between the variational latent representations is measured by

$$P(A|Z^*) = \prod_{i=1}^{2m} \prod_{j=1}^{2m} p(A_{ij}|\mathbf{z}_i^*, \mathbf{z}_j^*), \tag{9}$$

$$p(A_{ij} = 1|\mathbf{z}_i^*, \mathbf{z}_j^*) = \varphi(\mathbf{z}_i^{*T} \mathbf{z}_j^*), \tag{10}$$

where $\varphi(\cdot)$ is an activation operation with logistic sigmoid function. The inner product directly measures the similarity between two variational latent variables (nodes) to match the original graph input. For simplicity, we adopt an L2 distance to measure the graph difference rather than the original maximum likelihood estimation $(\min \log P(A|Z^*))$ [23]. This is represented by

$$\mathcal{L}_g = \sum\nolimits_{\forall A_{ij} > 0} ||A_{ij} - \varphi(\mathbf{z}_i^{*T}.\mathbf{z}_j^*)||_2^2. \tag{11}$$

Distribution Alignment. It aligns the intra-instance variance $p(Z^*)$ with the isotropic centered Gaussian with Kullback–Leibler divergence $p(Z^*|X, A) = \mathcal{N}(Z^*|Z, \boldsymbol{\sigma}^2)$, which is formulated by

$$\begin{aligned} \mathcal{L}_{kl} &= -KL(p(Z^*|X, A)||p(Z^*)) \\ &= -\frac{1}{4m} \sum_{\forall i,j \in \mathcal{B}} (1 + 2\log(\boldsymbol{\sigma}_i^{(j)}) - (\mathbf{z}_i^{(j)})^2 - (\boldsymbol{\sigma}_i^{(j)})^2). \end{aligned} \tag{12}$$

3.5. Joint Training

The overall learning objective function $\mathcal{L}$ is a combination of three components, formulated by

$$\mathcal{L} = \mathcal{L}_z + \mathcal{L}_r + \lambda \cdot \mathcal{L}_s. \tag{13}$$

λ is a weighting factor of the structural loss. A step-by-step illustration of PSLR is shown in Fig. 3: 1) The instance features are first extracted by the network using data *augmentations*; 2) The graph *latent* representation is calculated within each training batch; 3) The network is optimized using the adaptable softmax embedding with *enlarged* positive/negative similarity between the latent representations; 4) The *variational* latent representation is reconstructed to enhance the robustness; and 5) The *structural* information is aligned to reinforce the discriminability.

Siamese Network Training. As shown in Fig. 2, PSLR is trained with a Siamese network to guarantee training efficiency. At each training step, m image instances are randomly sampled and two random data augmentations are performed, then totally $2m$ images are fed into the network for training. The strategy avoids duplicated pairwise similarity measurement in Eq. 3 and 4, resulting in higher efficiency.

4. Experimental Results

We evaluate PSLR under two different settings: *Seen Testing Category* (CIFAR-10 [26] and STL-10 [5] datasets in § 4.1) and *Unseen Testing Category* (CUB200 [46], Car196 [25] and Product [35] datasets in § 4.2). In the former setting, training and testing sets share the same categories (*kNN classification protocol*), while in the second setting, they do not share any common categories (*zero-shot image retrieval protocol*).

4.1. Experiments on Seen Testing Categories

This subsection evaluates the learned embedding, where the testing samples share the same categories as training samples. Following [51, 56], we conduct the experiments on CIFAR-10 [26] and STL-10 [5] datasets, using the ResNet18 network [15] as the backbone. We fix the dimensions of the output feature embedding and latent representation to 128. We set the initial learning rate to 0.03, and then decay by 0.1 every 40 epochs after the first 120 epochs, for a total of 200 training epochs. To avoid trivial solutions, we use $\mathbb{I}_{2m}$ as the adjacent matrix A, and we may investigate a better graph construction strategy in the future. We set the temperature parameter τ to 0.1, the adaptable indicator η as 100 and $\lambda = 0.1$. We fix the batch size to 128 for all the comparison. PSLR is implemented on PyTorch, and optimized by SGD, where the weight decay parameter is 5×10^{-4} and momentum is 0.9. For data augmentation, *RandomResizedCrop*, *RandomGrayscale*, *ColorJitter*, and *RandomHorizontalFlip*) are adopted [56].

The weighted kNN classifier is adopted to evaluate the top-1 classification accuracy. The kNN classifier measures the visual similarity between learned features. Given a testing sample, we retrieve its top-k ($k = 200$ by default) nearest neighbors with the cosine similarity, and weighted voting is used to predict the label of the input testing sample.

Table 1: kNN accuracy (%) with different k on CIFAR-10 dataset.

Methods	k=5	k=20	k=200
RandomCNN	32.4	34.8	33.4
DeepCluster (1000) [4]	66.5	67.4	67.6
Exemplar [9]	73.2	74.0	74.5
NPSoftmax [51]	79.6	80.5	80.8
NCE [51]	79.4	80.2	80.4
ISIF [56]	82.4	83.1	83.6
AND† [19] (2 round)	82.7	83.6	84.2
AND† [19] (5 round)	84.8	85.9	86.3
AET‡ [57]	77.6	76.3	78.2
AVT‡ [37]	78.4	78.5	79.0
PSLR (1 round)	83.8	84.7	85.2
PSLR + AND (5 round)	**87.4**	**88.1**	**88.4**

† AND [19] is built with gradually neighborhood discovery and each round takes 200 epochs. Other methods are reported with 200 epochs.

‡ The results (AET [57] and AVT [37]) are obtained with features from the second convolutional block, while the last embedding layer does not preserve visual meaning and the accuracy for kNN search is very low.

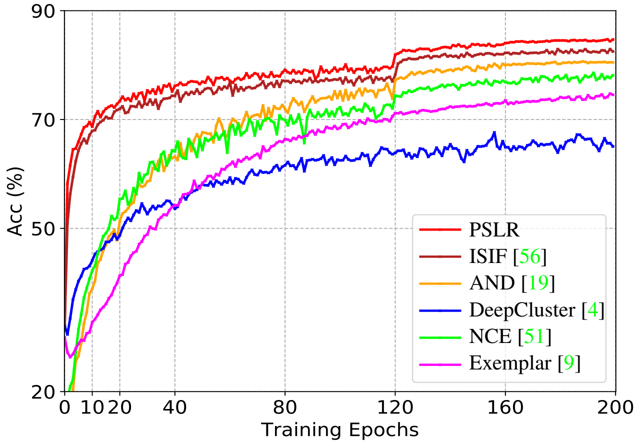


Figure 4: Learning curves on CIFAR-10 dataset. kNN accuracy (%) at each training epoch is reported. ($k = 200$)

4.1.1 CIFAR-10 Dataset

CIFAR-10 [26] contains 50,000/10,000 images (32 × 32) from the same 10 classes. We compare PSLR with eight unsupervised learning methods, as shown in Table 1. Note that ISIF [56] and AND [19] represent the state-of-the-art UEL methods, while AET [57] and AVT [37] indicates the state-of-the-art for UFL tasks, which learns linear classifiers with annotated labels using the unsupervisedly learned representations. The learning curves are shown in Fig. 4.

kNN Classification Accuracy. Table 1 demonstrates that PSLR achieves the best performance (85.2%) with 200 training epochs. Note that AND [19] achieves slightly better accuracy (86.3%) by continuously neighborhood mining after 1000 training epochs, while PSLR achieves 88.4% under this setting. Compared to ISIF [56] with softmax embedding on instance features, PSLR consistently improves the performance by optimizing the latent representation with structural information mining. The main reason is that learning with latent representation results in better

Table 2: Classification accuracy (%) with linear classifier (*Linear*) and kNN classifier (*kNN*) on STL-10 dataset.

Methods	Training	Linear	kNN
RandomCNN	None	-	22.4
k-MeansNet* [6]	105K	60.1	-
HMP* [2]	105K	64.5	-
Satck* [59]	105K	74.3	-
Exemplar* [9]	105K	75.4	-
NPSoftmax [51]	5K	62.3	66.8
NCE [51]	5K	61.9	66.2
DeepCluster(100) [4]	5K	56.5	61.2
ISIF [56]	5K	69.5	74.1
AND [19]	105K	76.8	80.2
ISIF [56]	105K	77.9	81.6
PSLR	105K	**78.8**	**83.2**

generalizability on testing samples as discussed in § 4.3. Compared to AET [57] and AVT [37]), they perform well in learning good "intermediate" features with additional linear classifier learning, but their performance for similarity-based search drops dramatically.

Efficiency. Fig. 4 illustrates the learning speed of all the comparison methods. We observe that both PSLR and ISIF [56] achieve much faster learning speed than the other competitors, by directly performing the optimization on features rather than the memory bank [19, 51]. This demonstrates that adaptable softmax embedding performed on the latent representation maintains the high efficiency and meanwhile improves the testing accuracy compared to [56].

4.1.2 STL-10 Dataset

STL-10 [5] is an image recognition dataset for unsupervised learning. It contains 5000 labeled images (96×96) from ten classes and 100,000 unlabeled images. We do not use the annotated labels for embedding learning. The testing set contains 8000 images from the same ten classes. We report the classification accuracy (%) using both the Linear Classifier (*Linear*) and kNN classifier (*kNN*) in Table 2. The linear classifier is trained with the 5000 labeled images based on the unsupervisedly learned features from 105K training images. We implement AND [19] under the same settings while other results are taken from [56].

Table 2 demonstrates that PSLR outperforms its counterparts under both evaluation metrics (kNN: 83.2%, Linear: 78.8%). When we use the 105K images for training, we achieve consistently better performance than the two state-of-the-art methods (AND [19] and ISIF [56]). Note that *Linear*, which measures the linear separability of the learned representations, requires additional classifier training with labeled images. In contrast, the kNN classifier directly measures the visual similarity with learned representations, which requires the similarity preservation between samples. In addition, this experiment also demonstrates that PSLR benefits from more training samples.

Table 3: Retrieval performance (%) on CUB200 dataset.

Methods	R@1	R@2	R@4	R@8
	Supervised Learning			
Lifted [35]	43.6	56.6	68.6	79.6
Clustering[41]	48.2	61.4	71.8	81.9
Triplet+ [14]	45.9	57.7	69.6	79.8
Smart+ [14]	49.8	62.3	74.1	83.3
N-pair [40]	45.4	58.4	69.5	79.5
	Unsupervised Learning			
Initial (FC)	39.2	52.1	66.1	78.2
Cyclic [28]	40.8	52.8	65.1	76.0
Exemplar [9]	38.2	50.3	62.8	75.0
NCE [51]	39.2	51.4	63.7	75.8
DeepCluster[4]	42.9	54.1	65.6	76.2
MOM [20]	45.3	57.8	68.6	78.4
AND [19]	47.3	59.4	71.0	81.0
ISIF [56]	46.2	59.0	70.1	80.2
PSLR	**48.1**	**60.1**	**71.8**	**81.6**

4.2. Experiments on Unseen Testing Categories

In this section, we conduct the experiments with unseen testing categories, where the training and testing categories do not overlap. We follow the settings described in [35, 56] and perform the experiments on three fine-grained image retrieval datasets, including *CUB200* [46], Stanford Online Product (*Product*) [35] and Car196 [25]. The fine-grained image classes make similarity mining extremely challenging since we do not use the semantic labels for training.

Datasets. *CUB200* [46] is a dataset with 200 bird species. The first 100 classes with 5,864 images are used for training, while the remaining 100 classes with 5,924 images for testing. Stanford Online Product (*Product*) [35] is an online product dataset with much more classes. 11,318 classes with a total of 59,551 images are used for training, while the remaining 11,316 classes with 60,502 images for testing. *Car196* [25] is a fine-grained car dataset. We use the first 98 classes with 8,054 images for training, while the rest 98 classes with 8,131 images for testing.

Implementation Details. We adopt the Inception-V1 network [42] pre-trained on ImageNet as our backbone following [20, 56]. A batch normalization layer followed by a fully connected layer (128-dim) is added after the pool5 layer. The feature dimension of the latent representation is set to 128. The initial learning rate is set to 0.001 without decay. The temperature parameter τ is set to 0.1 and η is set to 1. Other parameters and settings are exactly the same as in §4.1 for optimization. The training batch size is set to 64. The input images are first resized to 256×256, and randomly cropped with random horizontal flipping to 227×227 images before being fed into the network.

Evaluation Metrics. In the testing phase, a single center-cropped image is adopted for feature extraction. Following existing works [35, 14], the retrieval performance (Rank-k accuracy) is reported with cosine similarity [56].

Table 4: Retrieval performance (%) on Car196 dataset.

Methods	R@1	R@2	R@4	R@8
Initial (FC)	35.1	47.4	60.0	72.0
Exemplar [9]	36.5	48.1	59.2	71.0
NCE [51]	37.5	48.7	59.8	71.5
DeepCluster[4]	32.6	43.8	57.0	69.5
MOM [20]	35.5	48.2	60.6	72.4
AND [19]	38.4	49.6	60.2	72.9
ISIF [56]	41.3	52.3	63.6	74.9
PSLR	**43.7**	**54.8**	**66.1**	**76.2**

Table 5: Retrieval performance (%) on *Product* dataset.

Methods	R@1	R@10	R@100
Initial (FC)	40.8	56.7	72.1
Exemplar [9]	45.0	60.3	75.2
NCE [51]	46.6	62.3	76.8
DeepCluster[4]	34.6	52.6	66.8
MOM [20]	43.3	57.2	73.2
AND [19]	47.4	62.6	77.1
ISIF [56]	48.9	64.0	78.0
PSLR	**51.1**	**66.5**	**79.8**

Comparison to the State-of-the-arts. We compare the state-of-the-art unsupervised learning methods, including Exemplar [9], NCE [51], DeepCluster [4], MOM [20], AND [19] and ISIF [56]. The results are shown in Table 3, 4 and 5, respectively. Most of these results are taken from [56]. We also implement the state-of-the-art AND [19] under the same settings for comparison. Note that the mined neighbors with AND contain large amount of false positives since the different classes are quite similar in this setting. We also compare some supervised learning methods on the CUB200 dataset, as shown in Table 3.

The major challenge under unseen testing categories is that these categories do not occur in the training set, which requires visual similarity mining rather than fitting to training samples. Results on the three fine-grained image recognition datasets demonstrate that the instance-wise representation learning models (NCE [51], AND [19], ISIF [56] and PSLR) usually perform better than the label-mining methods (DeepCluster [4], MOM [20]). The main reason is that instance-wise supervision avoids wrong label estimation, making it more suitable for unsupervised learning under these fine-grained settings. AND [19] performs well on the CUB200 dataset when the neighborhood discovery is reliable with a good initialization model, but the performance drops dramatically when applied to *Car196* and *Product*, in which it is quite difficult to mine reliable neighborhood information. In comparison, PSLR does not rely on the initialized representation. Compared to ISIF [56], PSLR is also the clear winner due to its optimization on the latent representation with structural information. Our design shows better generalizability on unseen testing categories as verified in § 4.3. Meanwhile, PSLR even achieves comparable performance to some supervised methods on CUB200.

Table 6: Results on *Product* dataset without pre-trained network.

Methods	R@1	R@10	R@100
Random	18.4	29.4	46.0
Exemplar [9]	31.5	46.7	64.2
NCE [51]	34.4	49.0	65.2
MOM [20]	16.3	27.6	44.5
AND [19]	36.4	52.8	67.2
ISIF [56]	39.7	54.9	71.0
PSLR ($\eta = 1$)	40.4	55.6	69.7
PSLR ($\eta = 10$)	**42.3**	**57.7**	**72.5**

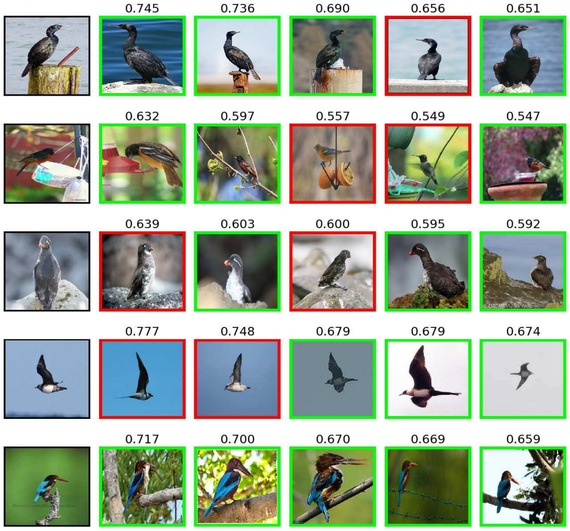


Figure 5: Retrieved examples with the calculated cosine similarities on the CUB200 dataset. The positive and negative retrieved results are framed in green and red, respectively.

Qualitative Results. To understand the learned embedding, we visualize some retrieved results on the CUB200 dataset as shown in Fig. 5. Although it contains some wrongly retrieved images with different semantic labels, most of the top-ranked images are visually similar to the query image. This demonstrates that PSLR can learn a good feature embedding to mine the underlying visual similarity. Interestingly, PSLR still obtains the correct results even when the bird images suffer from flipping variations (first and second row in Fig. 5). The main reason is that PSLR learns the data augmentation invariant features with random flipping, achieved by the latent representation learning.

Training from Scratch. We also evaluate PSLR on the large-scale Product dataset without using pre-trained ImageNet model (ResNet18) for initialization. The results of different methods are shown in Table 6. We observe that PSLR is again the clear winner, even without pre-trained model. The main reason is that using the randomly augmented samples as positives provides reliable positive supervision for unsupervised embedding learning. In comparison, MOM [20] and AND [19] suffer in this experiment due to the incorrect label mining with random initialized network for fine-grained categories.

4.3. Further Analysis

Effectiveness of Each Component. We evaluate each component in our proposed PSLR on the *Car196* dataset, as shown in Table 7. We observe that all the designed components contribute to the performance gain. The smooth variational reconstruction loss $\mathcal{L}_r$ enhances the robustness against signal noise and enriches the positive supervision, thus improving the testing performance on unseen categories. In addition, the graph structural preserving loss $\mathcal{L}_s$ facilitates the representation learning by mining the relationship cues among different instance representations.

Table 7: Effects of each component on Car196 dataset. Rank-k accuracy (%) is reported.

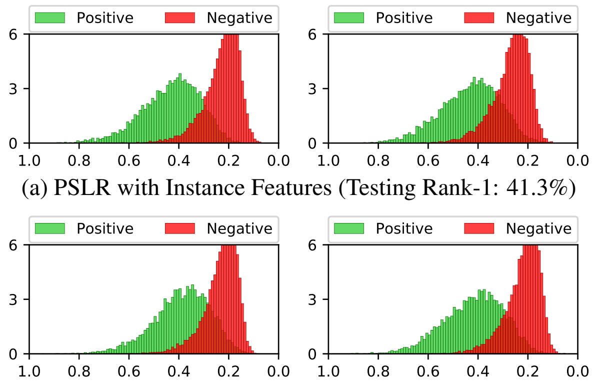
Strategies	R@1	R@2	R@4	R@8
$\mathcal{L}_z$	42.1	53.2	64.6	75.2
$\mathcal{L}_z + \mathcal{L}_r$	43.2	54.6	65.6	75.9
$\mathcal{L}_z + \mathcal{L}_r + \mathcal{L}_s$	43.7	54.8	66.1	76.2

Why Latent Representation Learning? We visualize the similarity distribution of the training and testing set on the *Car196* dataset. We calculate the cosine similarity distributions between the query features and their 5NN features from the same category (*Positive*) as well as 5NN features from different categories (*Negative*). The distributions of PSLR with latent representations or directly with instance features learning are shown in Fig. 6. Note that the latter directly optimizes Eq. 5 with the instance features.

Naturally, a more separated distribution (*Positive* vs. *Negative*) indicates a better feature embedding. Fig. 6 demonstrates that instance feature optimization achieves better separation than latent representation optimization on the training set but it performs worse on the testing set. This experiment shows that latent representation learning is more suitable for UEL with unseen testing categories, since it prevents over-fitting to the training instance features.

Why Adaptable Softmax? This subsection evaluates the adaptable softmax embedding on latent representation in §3.2. We plot the performance with different η (1, 10, 100) on one seen testing category dataset (CIFAR-10) and one unseen testing category dataset (*Product*), as shown in Fig. 7. We also report the performance of ISIF [56] with different η. Note that our proposed adaptable softmax with the adaptable factor η is equivalent to ISIF when $\eta = 1$.

We draw two conclusions from Fig. 7: 1) The adaptable factor significantly improves the performance for both PSLR and ISIF under two different settings. The main reason is that the probability difference between the positive and negative is enlarged with a large η. This enhances the discriminability with the enlarged negative samples, which shares similar spirit with hard negative sample mining [14]. 2) The proposed PSLR consistently outperforms its major counterpart ISIF in all the settings. This further demonstrates the superiority of latent representation learning.



(a) PSLR with Instance Features (Testing Rank-1: 41.3%)

(b) PSLR with Latent Representation (Testing Rank-1: 43.7%)

Figure 6: The cosine similarity distribution of the training (left column)) and testing (right column) sets from the *Car196* dataset.

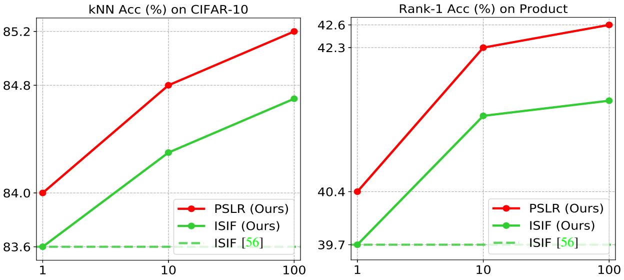


Figure 7: Results of different η in PSLR on CIFAR-10 and *Product* datasets with ResNet18 (training from scratch) to show the effectiveness of the adaptable softmax embedding. We show that ISIF [56] is also improved with the adaptable factor.

Backbone Network. In this part, we evaluate the generalizability of PSLR using different backbone networks. We further test ResNet18 and ResNet50 [15] on three fine-grained image recognition datasets with unseen testing categories. The rank-1 accuracy in Table 8 demonstrates that PSLR benefits from stronger backbone network structures.

Table 8: Rank-1 accuracy (%) with different backbone networks.

Backbone	CUB200	Car196	Product
InceptionV1	48.1	43.7	51.1
ResNet18	48.9	39.2	52.2
ResNet50	49.0	42.8	61.6

5. Conclusion

This paper presents a novel probabilistic structural latent representation (PSLR) for unsupervised embedding learning. We propose an adaptable softmax embedding to optimize the graph latent representation, which achieves superior performance and achieves high efficiency on both seen and unseen categories. Meanwhile, a smooth variational reconstruction loss is introduced to enhance the robustness against signal noise and enrich the positive supervision. A structural preserving loss is also developed to fully exploit the underlying relationship among different instances. Extensive experiments on five different datasets with cosine similarity have validated the effectiveness.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, pages 37–45, 2015. 2

[2] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013. 6

[3] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, pages 517–526, 2017. 2

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 1, 2, 5, 6, 7

[5] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223, 2011. 5, 6

[6] Adam Coates and Andrew Y Ng. Selecting receptive fields in deep networks. In *NIPS*, pages 2528–2536, 2011. 6

[7] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 1, 2

[8] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[9] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE TPAMI*, 38(9):1734–1747, 2016. 2, 5, 6, 7

[10] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, pages 766–774, 2014. 2

[11] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 2

[12] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 1

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 2

[14] Ben Harwood, BG Kumar, Gustavo Carneiro, Ian Reid, Tom Drummond, et al. Smart mining for deep metric learning. In *ICCV*, pages 2821–2829, 2017. 6, 8

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5, 8

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[17] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 4

[18] Fu Jie Huang, Y-Lan Boureau, Yann LeCun, et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, pages 1–8, 2007. 2

[19] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *ICML*, pages 2849–2858, 2019. 2, 3, 5, 6, 7

[20] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Mining on manifolds: Metric learning without labels. In *CVPR*, pages 7642–7651, 2018. 1, 2, 6, 7

[21] Xu Ji, Joao F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, pages 9865–9874, 2019. 2

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2013. 4

[23] Thomas N Kipf and Max Welling. Variational graph autoencoders. *arXiv preprint arXiv:1611.07308*, 2016. 2, 4

[24] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *CVPR*, pages 1920–1929, 2019. 2

[25] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013. 5, 6

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[27] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, pages 609–616, 2009. 2

[28] Dong Li, Wei-Chih Hung, Jia-Bin Huang, Shengjin Wang, Narendra Ahuja, and Ming-Hsuan Yang. Unsupervised visual representation learning by graph-based consistent constraints. In *ECCV*, pages 678–694, 2016. 6

[29] Tao Li, Zhiyuan Liang, Sanyuan Zhao, Jianhao Gong, and Jianbing Shen. Self-learning with rectification strategy for human parsing. In *CVPR*, 2020. 1

[30] Renjie Liao, Alex Schwing, Richard Zemel, and Raquel Urtasun. Learning deep parsimonious representations. In *NIPS*, pages 5076–5084, 2016. 1, 2

[31] Xudong Lin, Yueqi Duan, Qiyuan Dong, , Jiwen Lu, and Jie Zhou. Deep variantional metric learning. In *ECCV*, pages 689–704, 2018. 2

[32] Chaochao Lu and Xiaoou Tang. Surpassing human-level face verification performance on lfw with gaussianface. In *AAAI*, pages 3811–3819, 2015. 1

[33] R Manmatha, Chao-Yuan Wu, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, pages 2859–2867, 2017. 1

[34] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84, 2016. 1, 2

[35] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016. 1, 2, 5, 6

[36] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2

[37] Guo-Jun Qi, Liheng Zhang, Chang Wen Chen, and Qi Tian. Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *ICCV*, pages 8130–8139, 2019. 1, 5, 6

[38] Mohammad Sabokrou, Mohammad Khalooei, and Ehsan Adeli. Self-supervised representation learning via neighborhood-relational encoding. In *ICCV*, pages 8010–8019, 2019. 2

[39] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *CVPR*, pages 8247–8255, 2019. 2

[40] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, pages 1857–1865, 2016. 6

[41] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *CVPR*, pages 2206–2214, 2017. 2, 6

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 6

[43] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Robust boltzmann machines for recognition and denoising. In *CVPR*, pages 2264–2271, 2012. 2

[44] Daniel Tarlow, Kevin Swersky, Laurent Charlin, Ilya Sutskever, and Rich Zemel. Stochastic k-neighborhood selection for supervised and unsupervised learning. In *ICML*, pages 199–207, 2013. 2

[45] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, 2008. 2

[46] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5, 6

[47] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802, 2015. 2

[48] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, pages 5022–5030, 2019. 1

[49] Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. Detecting overfitting of deep generative networks via latent recovery. In *CVPR*, pages 11273–11282, 2019. 4

[50] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L. Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *CVPR*, pages 1910–1919, 2019. 1, 2

[51] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 2, 3, 5, 6, 7

[52] Mang Ye, Xiangyuan Lan, and Pong C. Yuen. Robust anchor embedding for unsupervised video person re-identification in the wild. In *ECCV*, 2018. 2

[53] Mang Ye, Jiawei Li, Andy J Ma, Liang Zheng, and Pong C Yuen. Dynamic graph co-matching for unsupervised video-based person re-identification. *IEEE TIP*, 28(6):2976–2990, 2019. 2

[54] Mang Ye, Andy J Ma, Liang Zheng, Jiawei Li, and Pong C Yuen. Dynamic label graph matching for unsupervised video re-identification. In *ICCV*, pages 5142–5150, 2017. 2

[55] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. Deep learning for person re-identification: A survey and outlook. *arXiv preprint arXiv:2001.04193*, 2020. 1

[56] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pages 6210–6219, 2019. 1, 2, 5, 6, 7, 8

[57] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *CVPR*, pages 2547–2555, 2019. 1, 2, 5, 6

[58] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666, 2016. 2

[59] Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann Lecun. Stacked what-where auto-encoders. *arXiv preprint arXiv:1506.02351*, 2015. 6

[60] Zhilin Zheng and Li Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In *CVPR*, pages 12192–12201, 2019. 2

[61] Tao Zhou, Huazhu Fu, Chen Gong, Jianbing Shen, Ling Shao, and Porikli Fatih. Multi-mutual consistency induced transfer subspace learning for human motion segmentation. In *CVPR*, 2020. 1

[62] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6002–6012, 2019. 2