Super-AND: A Holistic Approach to Unsupervised Embedding Learning

Anonymous authors
Paper under double-blind review

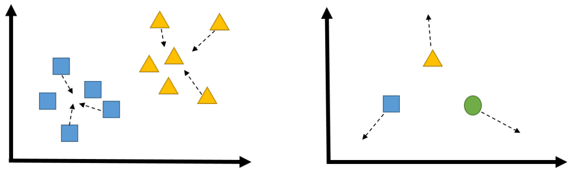
Abstract

Unsupervised embedding learning aims to extract good representations from data without the use of human-annotated labels. Such techniques are apparently in the limelight because of the challenges in collecting massive-scale labels required for supervised learning. This paper proposes a comprehensive approach, called Super-AND, which is based on the Anchor Neighbourhood Discovery model (Huang et al., 2019). Multiple losses defined in Super-AND make similar samples gather even within a low-density space and keep features invariant against augmentation. As a result, our model outperforms existing approaches in various benchmark datasets and achieves an accuracy of 89.2% in CIFAR-10 with the ResNet18 backbone network, a 2.9% gain over the state-of-the-art.

1 Introduction

Deep learning and convolutional neural network have become an indispensable technique in computer vision (LeCun et al., 2015; Krizhevsky et al., 2012; Lawrence et al., 1997). Remarkable developments, in particular, were led by supervised learning that requires thousands or more labeled data. However, high annotation costs have become a significant drawback in training a scalable and practical model in many domains. In contrast, unsupervised deep learning that requires no label has recently started to get attention in computer vision tasks. From clustering analysis (Caron et al., 2018; Ji et al., 2018), and self-supervised model (Gidaris et al., 2018; Bojanowski & Joulin, 2017) to generative model (Goodfellow et al., 2014; Kingma & Welling, 2013; Radford et al., 2016), various learning methods came out and showed possibilities and prospects.

Unsupervised embedding learning aims to extract visually meaningful representations without any label information. Here "visually meaningful" refers to finding features that satisfy two traits: (*i*) positive attention and (*ii*) negative separation (Ye et al., 2019; Zhang et al., 2017c; Oh Song et al., 2016). Data samples from the same ground truth class, i.e., positive samples, should be close in the embedding space (Fig. 1a); whereas those from different classes, i.e., negative samples, should be pushed far away in the embedding space (Fig. 1b). However, in the setting of unsupervised learning, a model cannot have knowledge about whether given data points are positive samples or negative samples.



(a) Positive attention (b) Negative separation

Figure 1: Illustration of two characteristics; (a) Positive attention, (b) Negative separation. Data points with same shape and same color in this plot are positive samples. Otherwise, they are negative samples.

Several new methods have been proposed to find 'visually meaningful' representations. The sample specificity method considers all data points as negative samples and separates them in the feature space (Wu et al., 2018; Bojanowski & Joulin, 2017). Although this method achieves high performance, its decisions are known to be biased from learning only from negative separation. One approach utilizes data augmentation to consider positive samples in training (Ye et al., 2019), which efficiently reduces any ambiguity in supervision while keeping invariant features in the embedding space. Another approach is called the Anchor Neighborhood Discovery (AND) model, which alleviates the complexity in boundaries by discovering the nearest neighbor among the data points (Huang et al., 2019). Each of these approaches overcomes different limitations of the sample specificity method. However, no unified approach has been proposed.

This paper presents a holistic method for unsupervised embedding learning, named Super-AND. Super-AND extends the AND algorithm and unifies various but dominant approaches in this domain with its unique architecture. Our proposed model not only focuses on learning distinctive features across neighborhoods, but also emphasizes edge information in embeddings and maintains the unchanging class information from the augmented data. Besides combining existing techniques, we newly introduce Unification Entropy loss (UE-loss), an adversary of sample specificity loss, which is able to gather similar data points within a low-density space. Extensive experiments are conducted on several benchmark datasets to verify the superiority of the model. The results show the synergetic advantages among modules of Super-AND. The main contributions of this paper are as follows:

- We effectively unify various techniques from state-of-the-art models and introduce a new loss, UE-loss, to make similar data samples gather in the low-density space.
- Super-AND outperforms all baselines in various benchmark datasets. It achieved an accuracy of 89.2% in the CIFAR-10 dataset with the ResNet18 backbone network, compared to the state-of-the-art that gained 86.3%.
- The extensive experiments and the ablation study show that every component in Super-AND contributes to the performance increase, and also indicate their synergies are critical.

Our model's outstanding performance is a step closer to the broader adoption of unsupervised techniques in computer vision tasks. The premise of data-less embedding learning is at its applicability to practical scenarios, where there exists only one or two examples per cluster. Codes and trained data for Super-AND are accessible via a GitHub link.[1]

2 Related Work

Existing research on unsupervised deep learning can be summarized into four groups below:

Generative model. This type of model is a powerful branch in unsupervised learning. By reconstructing the underlying data distribution, a model can generate new data points as well as features from images without labels. Generative adversarial network (Goodfellow et al., 2014) has led to rapid progress in image generation problems (Zhang et al., 2019; Arjovsky et al., 2017). While some attempts have been made in terms of unsupervised embedding learning (Radford et al., 2016), the main objective of generative models lies at mimicking the true distribution of each class, rather than discovering distinctive categorical information the data contains.

Self-supervised learning. This type of learning uses inherent structures in images as pseudo-labels and exploits labels for back-propagation. For example, a model can be trained to create embeddings by predicting the relative position of a pixel from other pixels (Doersch et al., 2015) or the degree of changes after rotating images (Gidaris et al., 2018). Predicting future frames of a video can benefit from this technique (Walker et al., 2016). Wu et al. (2018) proposed the sample specificity method that learns feature representation from capturing apparent discriminability among instances. All of these methods are suitable for unsupervised embedding learning, although there exists a risk of false knowledge from generated labels that weakly correlate with the underlying class information.

Learning invariants from augmentation. Data augmentation is a strategy that enables a model to learn from datasets with an increased variety of instances. Popular techniques include flipping, scaling, rotation, and grey-scaling. These techniques do not deform any crucial features of data, but only change the style of images. Some studies hence use augmentation techniques and train models

[1] (Anonymized GitHub) https://github.com/super-AND/super-AND

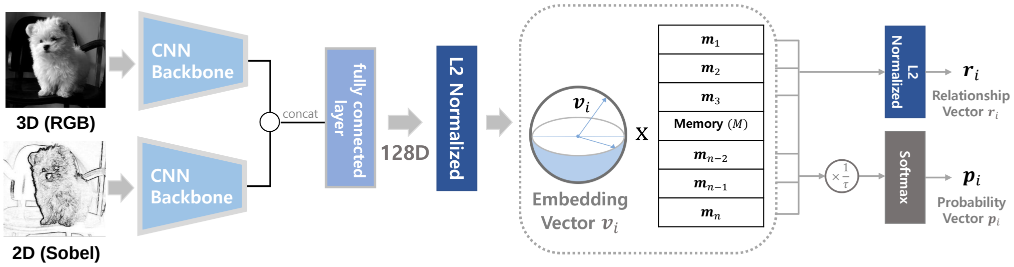


Figure 2: Illustration of basic architecture in Super-AND. Original RGB images and sobel-processed images are encoded by backbone CNN networks and concatenated. These vectors then projected to 128 dimensional sphere embedding with l2 normalization. Finally, adjacency relationship vector and probability vector are calculated.

to learn invariant features. In particular, Ji et al. (2018) used mutual information to extract invariant features between the augmented images, and Ye et al. (2019) regarded the augmented images as positive samples of original pictures for unsupervised feature learning. This study also adopts the same concept to reduce the distance of relationship vectors between the original and the augmented images.

Clustering analysis. This type of analysis is an extensively studied area in unsupervised learning, whose main objective is to group similar objects into the same class. Many studies either leveraged deep learning for dimensionality reduction before clustering (Schroff et al., 2015; Baldi, 2012) or trained models in an end-to-end fashion (Xie et al., 2016; Yang et al., 2016). Caron et al. (2018) proposed a concept called deep cluster, an iterative method that updates its weights by predicting cluster assignments as pseudo-labels. However, directly reasoning the global structures without any label is error-prone. The AND model, which we extend in this work, combines the advantages of sample specificity and clustering strategy to mitigate the noisy supervision via neighborhood analysis (Huang et al., 2019).

3 The Proposed Super-AND Model

Problem definition. Assume that there is an unlabeled image set $\mathcal{I}$, and a batch set $\mathcal{B}$ with n images: $\mathcal{B} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \subset \mathcal{I}$. Our goal is to get a feature extractor f_θ whose representations (i.e., $\mathbf{v}_i = f_\theta(\mathbf{x}_i)$) are "visually meaningful," a definition we discussed earlier.

Let $\hat{\mathcal{B}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ..., \hat{\mathbf{x}}_n\}$ be the augmentation set of input batches $\mathcal{B}$. Super-AND projects images $\mathbf{x}_i$, $\hat{\mathbf{x}}_i$ from batches $\mathcal{B}$, $\hat{\mathcal{B}}$ to 128 dimensional embeddings $\mathbf{v}_i$, $\hat{\mathbf{v}}_i$. During this process, the Sobel-processed images (Maini & Aggarwal, 2008) are also used, and feature vectors from both images are concatenated to emphasize edge information in embeddings (see the left side in Fig. 2). Then, the model computes the probability of images $\mathbf{p}_i$, $\hat{\mathbf{p}}_i$ being recognized as its own class with a non-parametric classifier (see the right side in Fig. 2). A temperature parameter ($\tau < 1$) was added to ensure a label distribution with low entropy (Hinton et al., 2015). To reduce the computational complexity in calculating embeddings from all images, we set up a memory bank M to save instance embeddings $\mathbf{m}_i$ accumulated from the previous steps, as similarly proposed by Wu et al. (2018). The memory bank M is updated by exponential moving average (Lucas & Saccucci, 1990). The probability vector $\mathbf{p}_i$ is defined in Eq 1, where the superscript j in vector notation (i.e., $\mathbf{v}^j$) represents the j-th component value of a given vector.

$$\mathbf{p}_i^j = \frac{\exp(\mathbf{m}_j^\top \mathbf{v}_i/\tau)}{\sum_{k=1}^{n} \exp(\mathbf{m}_k^\top \mathbf{v}_i/\tau)} \tag{1}$$

$$\mathbf{r}_i = \frac{\mathbf{v}_i \cdot \mathbf{m}_i}{||\mathbf{v}_i \cdot \mathbf{m}_i||_2} \tag{2}$$

We define the neighborhood relationship vectors $\mathbf{r}_i$, $\hat{\mathbf{r}_i}$, and compute these vectors by the cosine similarity between the embedding vectors $\mathbf{v}_i$, $\hat{\mathbf{v}}_i$, and the memory bank M (Eq 2). The extracted vectors are used to define the loss term that detects a discrepancy between neighborhoods. The loss term also enforces features $\mathbf{v}$ to remain unchanged even after data augmentation. The loss term is written as

$$L_{and} = AND(M, P, \mathcal{N}) \tag{3}$$

$$L_{ue} = UELoss(M, V) \tag{4}$$

$$L_{aug} = AugLoss(R, \hat{R}) \tag{5}$$

$$L_{total} = L_{and} + w(t) \times L_{ue} + L_{aug} \tag{6}$$

where $\mathcal{N}$ is the set of progressively selected pairs discovered by the nearest neighbor algorithm, and $V, R, \hat{R}, P$ are matrices of concatenated embedded vectors $\mathbf{v}_i, \mathbf{r}_i, \hat{\mathbf{r}}_i, \mathbf{p}_i$ from the batch image set, respectively. $w(t)$ is the hyper-parameter that controls the weights of UE-loss. The algorithm below describes how to train Super-AND.

Algorithm 1: Main algorithm for training Super-AND.

```
Input  : Unlabeled image set $\mathcal{I}$, encoder $f_\theta$ to train, the number of total rounds for training:
         Rounds, and the number of total epochs for training: Epochs
Output: Trained encoder $f_\theta$
Memory $M$ initialized
for $r \leftarrow 1$ to Rounds do
    $\tilde{\mathcal{N}} \leftarrow$ NeighborhoodDiscovery($\mathcal{I}, f_\theta$)
    $\mathcal{N} \leftarrow$ NeighborhoodSelection($\tilde{\mathcal{N}}, f_\theta, r$)
    for $t \leftarrow 1$ to Epochs do
        Get batch $\mathcal{B}$ from $\mathcal{I}$
        $\hat{\mathcal{B}} \leftarrow$ DataAugmentation($\mathcal{B}$)
        $V,\ \hat{V} \leftarrow f_\theta(\mathcal{B}),\ f_\theta(\hat{\mathcal{B}})$
        $R, \hat{R} \leftarrow$ CosineSimilarity($M, V$), CosineSimilarity($M, \hat{V}$)
        $L_{and},\ L_{ue},\ L_{aug} \leftarrow AND(M, P, \mathcal{N}),\ UELoss(M, V),\ AugLoss(R, \hat{R})$
        $L_{total} \leftarrow L_{and} + w(t) \times L_{ue} + L_{aug}$
        $M \leftarrow$ MemoryUpdate($M$)
        Compute gradient and update weights by backpropagation
    end
end
```

3.1 Neighborhood discovery and supervision

Existing clustering methods like (Caron et al., 2018; Xie et al., 2016) train networks to find an optimal mapping. However, their learned decisions are unstable due to initial randomness, and some overfitting can occur during the training period (Zhang et al., 2017a). To tackle these limitations, the AND model suggests a finer-grained clustering focusing on 'neighborhoods'. By regarding the nearest-neighbor pairs as local classes, AND can separate data points that belong to different neighborhood sets from those in the same neighborhood set. We adopt this neighborhood discovery strategy in our Super-AND.

The AND algorithm has three main steps: (1) neighborhood discovery, (2) progressive neighborhood selection with curriculum, and (3) neighborhood supervision. For the first step, the k nearest neighborhood (k-NN) algorithm is used to discover all neighborhood pairs (Eq 7 and Eq 8), and these pairs are progressively selected for curriculum learning. We choose a small part of neighborhood pairs at the first round, and gradually increase the amount of selection for training (*Current/Total rounds* $\times$ 100%). Since we cannot assure that every neighborhood is visually similar, this progressive method helps provide a consistent view of local class information for training at each round.

$$\tilde{\mathcal{N}}(\mathbf{x}) = \{\mathbf{x}_i | \mathbf{x}_i \neq \mathbf{x},\ f_\theta(\mathbf{x}_i)^\top f_\theta(\mathbf{x}) \text{ is top-1 in } \mathcal{I}\} \tag{7}$$

$$\tilde{\mathcal{N}} = \{(\mathbf{x}_i, \tilde{\mathcal{N}}(\mathbf{x}_i)) \mid \mathbf{x}_i \in \mathcal{I}\} \tag{8}$$

When selecting candidate neighborhoods for local classes, the entropy of probability vector $H(\mathbf{x}_i)$ is utilized as a criterion (Eq 9). Probability vector $\mathbf{p}_i$, obtained from softmax function (Eq 1), shows the visual similarity between training instances in a probabilistic manner. Data points with low entropy represent they reside in a relatively low-density area and have only a few surrounding neighbors. Neighborhood pairs containing such data points likely share consistent and easily distinguishable features from other pairs. We select neighborhood set $\mathcal{N}$ from $\tilde{\mathcal{N}}$ that is in a lower entropy order.

$$H(\mathbf{x}_i) = -\sum_j \mathbf{p}_i^j \log \mathbf{p}_i^j \tag{9}$$

The AND-loss function is defined to distinguish neighborhood pairs from one another. Data points from the same neighborhoods need to be classified in the same class (i.e., the left-hand term in Eq 10). If any data point is present in the selected pair, it is considered to form an independent class (i.e., the right-hand term in Eq 10).

$$L_{and} = -\sum_{i\in(\mathcal{B}\cap\mathcal{N})} \log\Big(\sum_{j\in\tilde{\mathcal{N}}(\mathbf{x}_i)} \mathbf{p}_i^j\Big) - \sum_{i\in(\mathcal{B}\cap\mathcal{N}^c)} \log(\mathbf{p}_i^i) \tag{10}$$

3.2 Unification entropy loss

Existing sample specificity methods (Wu et al., 2018; Bojanowski & Joulin, 2017) consider every single data point as a prototype for a class. They use the cross-entropy loss to separate all data points in the L2-normalized embedding space. Due to its confined space by normalization, data points cannot be placed far away from one another, and this space limitation induces an effect that leads to a concentration of positive samples, as shown in Fig. 1a.

The unification entropy loss (UE-loss) is able to even strengthen the concentration-effect above. We define the UE-loss as the entropy of the probability vector $\tilde{\mathbf{p}}_i$. Probability vector $\tilde{\mathbf{p}}_i$ is calculated from the softmax function and represents the similarity between instances except for instance itself (Eq 11). By excluding the class of one's own, minimizing the loss makes nearby data points attract each other — a concept that is contrary to minimizing the sample specificity loss. Employing both AND-loss and the UE-loss will enforce similar neighborhoods to be positioned close while keeping the overall neighborhoods as separated as possible. This loss is calculated as in Eq 12.

$$\tilde{\mathbf{p}}_i^j = \frac{\exp(\mathbf{m}_j^\top \mathbf{v}_i/\tau)}{\sum_{k=1,k\neq i}^n \exp(\mathbf{m}_k^\top \mathbf{v}_i/\tau)} \qquad \tilde{H}(\mathbf{x}_i) = -\sum_{j\neq i} \tilde{\mathbf{p}}_i^j \log \tilde{\mathbf{p}}_i^j \tag{11}$$

$$L_{ue} = -\sum_i \tilde{H}(\mathbf{x}_i) \tag{12}$$

3.3 Learning invariants in data augmentation

Unsupervised embedding learning aims at training encoders to extract visually meaningful features that are consistent with ground truth labels. Such learning cannot use any external guidance on features. Several previous studies tried to infer which features are substantial in a roundabout way; data augmentation is one such solution (Ye et al., 2019; Ji et al., 2018; Perez & Wang, 2017; Volpi et al., 2018). Since augmentation does not deform the underlying data characteristics, invariant features learned from the augmented data will still contain the class-related information. Naturally, a training network based on these features will show performance gain.

We define the Augmentation-loss to learn invariant image features. Assume that there is an image along with its augmented versions. We may regard every augmentation instance as a positive sample. The neighborhood relationship vectors, which show the similarity between all instances stored in memory, should also be similar to initial data points than other instances in the same batch. In Eq 13, the probability of an augmented instance that is correctly identified as class-i is denoted as $\bar{\mathbf{p}}_i^i$; and that of i-th original instance that is wrongly identified as class-j ($j \neq i$), $\bar{\mathbf{p}}_i^j$. The Augmentation-loss is then defined to minimize misclassification over instances in all batches (Eq 14).

$$\bar{\mathbf{p}}_i^i = \frac{\exp(\mathbf{r}_i^\top \hat{\mathbf{r}}_i/\tau)}{\sum_{k=1}^n \exp(\mathbf{r}_k^\top \hat{\mathbf{r}}_i/\tau)} \qquad \bar{\mathbf{p}}_i^j = \frac{\exp(\mathbf{r}_j^\top \mathbf{r}_i/\tau)}{\sum_{k=1}^n \exp(\mathbf{r}_k^\top \mathbf{r}_i/\tau)},\ \ j \neq i \tag{13}$$

$$L_{aug} = -\sum_i \sum_{j\neq i} \log(1 - \bar{\mathbf{p}}_i^j) - \sum_i \log(\bar{\mathbf{p}}_i^i) \tag{14}$$

4 EXPERIMENTS

The evaluation involved extensive experiments. We enumerated the model with different backbone networks on two kinds of benchmarks: coarse-grained and fine-grained. Our ablation study helps speculate which components of the model are critical in performance. Finally, the proposed model is compared to the original AND from different perspectives.

4.1 IMPLEMENTATION DETAILS

Datasets. A total of six image datasets are utilized, where three are coarse-grained datasets: (1) ***CIFAR-10*** (Krizhevsky et al., 2009): CIFAR-10 dataset has 10 classes images with 32 × 32 pixels. (2) ***CIFAR-100***: CIFAR-100 consists of the same images in CIFAR-10, but it has 100 classes. (3) ***SVHN*** (Netzer et al., 2011): Street View House Numbers (SVHN) is the real-world dataset with 10 classes of digit images of 32×32 pixels. Fine-grained datasets include: (4) ***Stanford Dogs*** (Khosla et al., 2011) contains 120 breeds of dog images, (5) ***CUB-200*** (Welinder et al., 2010): this Caltech-UCSD Birds dataset contains 200 species of bird images, and (6) ***STL-10*** (Coates et al., 2011) contains images in 96×96 pixels of 10 classes such as airplanes and birds. Dataset (6) is used for qualitative analysis.

Training. We used AlexNet (Krizhevsky et al., 2012) and ResNet18 (He et al., 2016) as the backbone networks. Hyper-parameters were tuned in the same way as the AND algorithm. We used SGD with Nesterov momentum 0.9 for the optimizer. We fixed the learning rate as 0.03 for the first 80 epochs, and scaled-down 0.1 every 40 epochs. The batch size is set as 128, and the model was trained in 5 rounds and 200 epochs per round. Weights for UE-loss $w(t)$ (Eq 6) are initialized from 0 and increased 0.2 every 80 epochs. For Augmentation-loss, we used four types: Resized Crop, Grayscale, ColorJitter, and Horizontal Flip. Horizontal Flip was not used in the case of the SVHN dataset because the SVHN dataset is digit images. Update momentum of the exponential moving average for memory bank was set to 0.5.

Evaluation. Following the method from Wu et al. (2018), we used the weighted k-NN classifier for making prediction. Top k-nearest neighbors $\mathcal{N}_{top}$ were retrieved and used to predict the final outcome in a weighted fashion. We set $k = 200$ and the weight function for each class c as $\sum_{i \in \mathcal{N}_{top}} \exp(\mathbf{v}_i^\top M/\tau) \cdot \mathbf{1}(c_i = c)$, where c_i is the class index for i-th instance. Top-1 classification accuracy was used for evaluation.

4.2 RESULTS & COMPONENT ANALYSIS

Baseline models. We adopt six state-of-the-art baselines for comparison. They are (1) *Split-Brain* (Zhang et al., 2017b), (2) *Counting* (Noroozi et al., 2017), (3) *DeepCluster* (Caron et al., 2018), (4) Instance (Wu et al., 2018), (5) *ISIF* (Ye et al., 2019), and (6) *AND* (Huang et al., 2019). For fair comparison, the same backbone networks were used.

Coarse-grained evaluation. Table 1 describes the object classification performance of seven models, including the proposed Super-AND on three coarse-grained datasets: CIFAR-10, CIFAR-100, and SVHN. Super-AND surpasses state-of-the-art baselines on all datasets except for one case, where the model underperforms marginally on CIFAR-100 with AlexNet. One notable observa-

Table 1: k-NN Evaluation on coarse-grained datasets. Results that are marked as * are borrowed from the previous works (Huang et al., 2019; Ye et al., 2019).

Dataset	CIFAR-10		CIFAR-100		SVHN	
Network	ResNet18	AlexNet	ResNet18	AlexNet	ResNet18	AlexNet
Split-Brain*	-	11.7	-	1.3	-	19.7
Counting*	-	41.7	-	15.9	-	43.4
DeepCluster*	67.6	62.3	-	22.7	-	84.9
Instance	80.8	60.3	50.7	32.7	93.6	79.8
ISIF	83.6	74.4	54.4	**44.1**	91.3	89.8
AND	86.3	74.8	57.2	41.5	94.4	90.9
Super-AND	**89.2**	**75.6**	**61.5**	42.7	**94.9**	**91.9**

Table 2: k-NN evaluation on fine-grained datasets. (SD: Stanford Dogs)

Dataset	SD	CUB-200
Instance	27.0	11.6
ISIF	31.4	13.2
AND	32.3	14.4
Super-AND	**39.0**	**17.6**

Table 3: Backbone network Analysis on CIFAR-10.

Network	Accuracy
AlexNet	75.6
ResNet18	89.2
ResNet101	90.5

Table 4: Ablation study on CIFAR-10.

Network	Accuracy
Full	89.2
w/o UE	88.7
w/o Sobel	88.3
w/o Aug	86.4

tion is that the difference between previous models and super-AND is mostly larger in the case of ResNet18 than the AlexNet backbone network. These results reveal that our model is superior to other methods and may indicate that our methodology can give more benefits to stronger CNN architectures.

Fine-grained evaluation. We perform evaluations on fine-grained datasets that require the ability to discriminate subtle differences between classes. Table 2 shows that Super-AND achieves an outstanding performance compared to three baselines with the ResNet18 backbone network. We excerpted the results of Instance and AND model from the previous work.

Backbone network. We tested the choice of backbone networks in terms of classification performance. AlexNet, ResNet18, and ResNet101 are used and evaluated on CIFAR-10, as shown in Table 3. From the results, we can infer that the stronger the backbone network our model has, the better the performance model can produce.

Ablation study. To verify every component does its role and has some contribution to the performance increase, an ablation study was conducted. Since Super-AND combines various mechanisms based on AND algorithm, we study the effect of removing each component: (1) Super-AND without the UE-loss, (2) Super-AND without the Sobel filter, (3) Super-AND without the Augmentation-loss. Table 4 displays the evaluation result based on the CIFAR-10 dataset and the ResNet18 backbone network. We found that every component contributes to the performance increase, and a particularly dramatic decrease in performance occurs when removing the Augmentation-loss.

Initialization. Instead of running the algorithm from an arbitrary random model, we can pre-train the network with "good" initial data points to discover consistent neighborhoods. We investigate two different initialization methods and check whether the choice is critical. Three models were compared: (1) a random model, (2) an initialized model with instance loss (Wu et al., 2018) from AND, and (3) an initialized model with multiple losses from Super-AND. Table 5 shows that the choice of initialization is not significant, and solely using the instance loss even has an adverse effect on performance. This finding implies that Super-AND is robust to random initial data points, yet the model will show an unexpected outcome if initialization uses ambiguous knowledge.

Table 5: Analysis of different initialization methods on CIFAR-10.

Initialization method	Random	Instance	Instance + UELoss + AugLoss
Accuracy	89.2	87.5	89.1

4.3 Comparison to the Anchor Neighborhood Discovery (AND) Model

Embedding quality analysis. Super-AND leverages the synergies from learning both similarities in neighborhoods and invariant features from data augmentation. Super-AND, therefore, has a high capability of discovering cluster relationships, compared to the original AND model that only uses the neighborhood information. Fig. 3 exploits t-SNE (Maaten & Hinton, 2008) to visualize the learned representations of three of the selected classes based on the two algorithms in CIFAR-10. The plot demonstrates that Super-AND discovers consistent and discriminative clusters.

We investigated the embedding quality by evaluating the class consistency of selected neighborhoods. Cheat labels are used to check whether neighborhood pairs come from the same class. Since both algorithms increase the selection ratio every round when gathering the part of discovered neighborhoods, the consistency of selected neighborhoods will naturally decrease. This relationship is

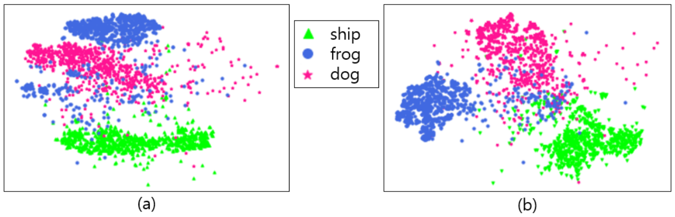


Figure 3: t-SNE visualization for the learned representations of three selected classes from CIFAR-10 in (a) AND (Huang et al., 2019) and (b) Super-AND.

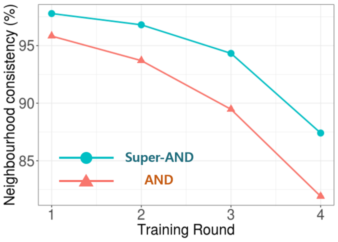


Figure 4: Neighborhood consistency over training rounds.

drawn in Fig. 4. The reduction for Super-AND, nonetheless, is not significant compared to AND: our model maintains high-performance throughout the training rounds.

Qualitative study. Fig. 5 illustrates the top-5 nearest retrievals of AND (i.e., upper rows) and Super-AND (i.e., lower rows) based on the STL-10 dataset. The example queries shown are dump trucks, airplanes, horses, and monkeys. Images with red frames, which indicate negative samples, appear more frequently for AND than Super-AND. This finding implies that Super-AND excels in capturing the class information compared to AND. Its clusters are robust to misleading color information and well recognize the shape of objects within images. For example, in the case of the airplane query, pictures retrieved from Super-AND are consistent in shape while AND results confuse a cruise picture as an airplane. The color composition in Super-AND is also more flexible and can find a red dump truck or a spotted horse, as shown in the examples.

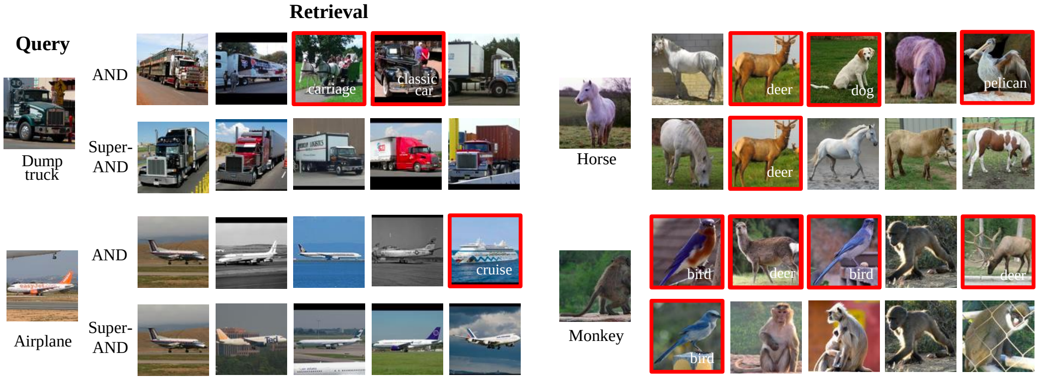


Figure 5: The nearest-neighbor retrievals of example queries from STL-10. The upper retrieval row from every query shows the results from the AND model, and the lower ones are from the Super-AND model. The left-side results are successful cases for both models, and the right-side results are failure cases. Images with surrounding red frames indicate the wrongly retrieved negative samples.

5 Conclusion

This paper presents Super-AND, a holistic technique for unsupervised embedding learning. Besides the synergetic advantage combining existing methods brings, the newly proposed UE-loss that groups nearby data points even in a low-density space while maintaining invariant features via data augmentation. The experiments with both coarse-grained and fine-grained datasets demonstrate our model's outstanding performance against the state-of-the-art models. Our efforts to advance unsupervised embedding learning directly benefit future applications that rely on various image clustering tasks. The high accuracy achieved by Super-AND makes the unsupervised learning approach an economically viable option where labels are costly to generate.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proc. of the International conference on machine learning*, pp. 214–223, 2017.

Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proc. of the ICML workshop on unsupervised and transfer learning*, pp. 37–49, 2012.

Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *Proc. of the International Conference on Machine Learning-Volume 70*, pp. 517–526. JMLR. org, 2017.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proc. of the European Conference on Computer Vision*, pp. 132–149, 2018.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proc. of the international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proc. of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. of the Advances in neural information processing systems*, pp. 2672–2680, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *Proc. of the International Conference on Machine Learning*, pp. 2849–2858, 2019.

Xu Ji, Joo F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. *arXiv preprint arXiv:1807.06653*, 2018.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *Proc. of the First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of the Advances in neural information processing systems*, pp. 1097–1105, 2012.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

James M Lucas and Michael S Saccucci. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, 32(1):1–12, 1990.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing*, 3(1), 2008.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proc. of the IEEE International Conference on Computer Vision*, pp. 5898–5906, 2017.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4004–4012, 2016.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. of the International Conference on Machine Learning*, 2016.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5495–5504, 2018.

Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Proc. of the European Conference on Computer Vision*, pp. 835–851, 2016.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proc. of the International conference on machine learning*, pp. 478–487, 2016.

Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156, 2016.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6210–6219, 2019.

Dejiao Zhang, Yifan Sun, Brian Eriksson, and Laura Balzano. Deep unsupervised clustering using mixture of autoencoders. *arXiv preprint arXiv:1712.07788*, 2017a.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proc. of the International Conference on Machine Learning*, pp. 7354–7363, 2019.

Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1058–1067, 2017b.

Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proc. of the IEEE International Conference on Computer Vision*, pp. 4595–4603, 2017c.