Convolutional Fine-Grained Classification With Self-Supervised Target Relation Regularization

Kangjun Liu, Ke Chen, Member, IEEE, and Kui Jia, Member, IEEE

Abstract—Fine-grained visual classification can be addressed by deep representation learning under supervision of manually pre-defined targets (e.g., one-hot or the Hadamard codes). Such target coding schemes are less flexible to model inter-class correlation and are sensitive to sparse and imbalanced data distribution as well. In light of this, this paper introduces a novel target coding scheme – dynamic target relation graphs (DTRG), which, as an auxiliary feature regularization, is a self-generated structural output to be mapped from input images. Specifically, online computation of class-level feature centers is designed to generate cross-category distance in the representation space, which can thus be depicted by a dynamic graph in a non-parametric manner. Explicitly minimizing intra-class feature variations anchored on those class-level centers can encourage learning of discriminative features. Moreover, owing to exploiting inter-class dependency, the proposed target graphs can alleviate data sparsity and imbalancedness in representation learning. Inspired by recent success of the mixup style data augmentation, this paper introduces randomness into soft construction of dynamic target relation graphs to further explore relation diversity of target classes. Experimental results can demonstrate the effectiveness of our method on a number of diverse benchmarks of multiple visual classification, especially achieving the state-of-the-art performance on three popular fine-grained object benchmarks and superior robustness against sparse and imbalanced data. Source codes are made publicly available at https://github.com/AkonLau/DTRG.

Index Terms—Fine-grained visual recognition, image classification, target codes, deep representation learning.

I. INTRODUCTION

THE problem of image classification [1], [2], [3] aims to categorize still images into semantic classes according to visual appearance of objects in images. Such a problem is made more challenging to recognize fine-grained objects, i.e., subclass of animal species such as birds [4] and dogs [5], or the brand and model of man-made products such as cars [6] and aircraft [7], which can be addressed by deep representation learning [8], [9] (consists of deep feature encoding and a typical classifier). The challenges of classifying fine-grained classes in deep feature encoding lie in coping with cross-class visual similarity and within-class variations intrinsically caused by object poses, in addition to other extrinsic imaging factors such as varying viewing angles and illumination.

Most of existing methods [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] focus on learning discriminative features to discover subtle appearance difference spatially distributed on local parts. A number of existing methods [10], [11], [12], [13], [14] attempt to locate and align pivotal parts of object by leveraging extra supervision of key points or bounding boxes of parts, which demand laborious annotations and are thus limited by the size of training data. More recent part-based methods [15], [16] and global feature encoding [18], [19], [20], [22], [23] can either discover distinguishable regions in a weakly-supervised learning manner, or enrich global representations with high-order information or instance-wise discrimination, to avoid utilizing extra part annotations.

The aforementioned methods are typically under supervision of one-hot target code [2], [3] or its soft variants such as label smoothing [24], [25]. Very few works including label relation graph [26] and the Hadamard code [27] attempt to exploit inter-class correlation to avoid inconsistent predictions and also transfer knowledge across classes. On one hand, the label relation graph can enrich the structure in the target space based on prior knowledge, which desires additional expensive professional efforts on annotating target relation. On the other hand, the target codes based on the Hadamard code aim to implicitly model inter-class relation using its redundant positive elements, whose optimization could be sub-optimal due to random assignment of target codes to semantic classes. More importantly, existing target coding schemes are manually pre-defined and thus less flexible to model complex relations between targets.

In light of this, a novel target relation scheme is proposed as an auxiliary feature regularization in a deep representation learning framework, which can discover latent correlation in the target space with no price of additional human efforts. Moreover, owing to mining target correlation, the proposed method can inherently mitigate negative effects of sparse and imbalanced data distribution, as samples from correlated classes can support to distinguish those from long-tailed classes. To this end, two key issues need to be addressed, i.e., discovering class-level representation within each fine-grained category and constructing target relation representation across classes.

To generate class-level centers in the feature space, existing parametric center generation method [28] learns to discover the...
class-level centers by constraining distance between samples’ features and their corresponding centers. Consequently, such a learnable parametric center generation can minimize the intra-class variance, but is not stable and robust against sparse and imbalanced data distribution. Such an observation inspires us to propose an online center generation method in a parameter-free fashion. We observe that for class-level representations (i.e., feature centers), which are the output of deep feature encoding, they are non-consistent when training epoch evolves. In this way, generation of dynamic class-level representations in our scheme is extreme simple, which are iteratively updated by the mean value of all samples’ features from the same class in the last epoch. Encouraged by the success of the center loss in [28], we adaptively propose an online center loss to minimize intra-class feature variations to improve discrimination of features. The proposed online center loss shares similar concept with the online label smoothing [25], but their goal is to generate soft labels revealing data distribution rather than capture inter-class target relation.

Given the class-level representations, we propose a novel dynamic target relation graph (DTRG) via measuring relative distance in the feature space to regularize deep feature encoding to incorporate latent inter-class relation as an extra similarity metric of feature encoding. As a result, samples from each class are not only supervised by the corresponding one-hot target code, but also constrained by all the other class-level centers in the feature space, which is also visualized in Fig. 1. Moreover, exploiting target relation in our scheme can also alleviate sparse and long tailed data distribution, which can further improve robustness of deep representations. Note that, the DTRG is non-parametric and can be readily inserted in any existing classification backbones for enhancing representation learning. To further expand the rich structure in the target space, recent soft-label based data augmentation such as vanilla Mixup [29], CutMix [30] and more recent SnapMix [22] can be employed. Experiments on several benchmarks including three popular fine-grained datasets can verify the effectiveness of our method, especially for a long-tail distributed data.

Main contributions can be summarized as follows:
1) We propose a novel non-parametric feature regularization to discover latent correlation across target classes in deep representation learning for fine-grained object classification, which is generic and could be readily adopted in existing classification methods.
2) Technically, we develop an online center loss to generate dynamic class-level representations, which are then utilized to construct a novel dynamic target relation graph. The proposed scheme cannot only minimize intra-class feature variance but also capture inter-class correlation to encourage discriminative features.
3) Diverse inter-class target relation in the proposed DTRG can be further boosted via semantically interpolation of the mixup-style data augmentation.
4) Our method can beat recent comparative methods on multiple benchmarks of different visual classification tasks, achieving the state-of-the-art accuracy on three popular fine-grained object benchmarks, and superior robustness against sparse and imbalanced data.

The remainder of the paper is organized as follows. We briefly describe related work in Sec. II, and detailed introduce our proposed graph similarity based feature regularization in a typical deep representation learning for fine-grained object classification in Sec. III. Experiments and evaluations on three widely-used datasets and some other datasets are presented in Sec. IV, then followed by our conclusion in Sec. V.

II. RELATED WORKS

Fine-grained visual classification has attracted wide attention in computer vision for decades, and a large number of algorithms have been proposed in the context of deep learning. Existing methods can be categorized into two main groups – part-based [10], [11], [12], [14], [15], [16], [17], [31], [32], [33], [34], [35], [36] and part-free [18], [19], [20], [21], [23], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48] which will be respectively investigated in details as follows.

A. Part-Based Methods

The pioneering methods [10], [11], [12], [14], [15] attempted to learn discriminative features by detecting and aligning the key parts of objects. Zhang et al. [11] proposed a bottom-up region proposal based method for part
location detection, which aims to learn a pose-normalized representation to discount negative effects of object poses. Huang et al. [15] proposed a Part-stacked Network for locating object parts by modeling the subtle differences of object parts. Ge et al. [49] built a complementary part detection model and then fused features of these different parts with a bi-direction LSTM network. Owing to its excellent performance for fine-grained features learning, the part-based methods [16], [49] remain active and recent concern is to spot discriminative parts without extra part annotations. Moreover, a number of attention-based methods [17], [31], [32], [33], [34], [35], [36], [50] are designed for detecting the discriminative regions and learning features by imitating the attention mechanism of humans. In [17] and [36], an attention module was designed to detect discriminative regions iteratively, from which fine-grained local features are extracted for object classification. Recently, inspired by the self-attention mechanism of vision transformer (ViT) [51], an amount of transformer-based algorithms have been proposed to extract discriminative features of fine-grained objects. He et al. [52] proposed a transformer-based method via selecting discriminative image patches with an attention map. Wang et al. [53] proposed a feature fusion vision transformer (FFVT) to enrich representations of fine-grained classes via aggregating local information across layers. Evidently, existing part-based methods aim to improve feature discrimination via enhancing distinguished details in local regions rather than exploiting inter-class relation as our method.

B. Part-Free Methods

An alternative group of fine-grained classifiers focus on global feature encoding, which can be further divided into two sub-categories – fusion-based [18], [19], [23], [37], [38], [39], [40], [41], [48] and regularization-based [46], [47].

1) Fusion-Based: Lin et al. [18] were the first to propose the bilinear models, which can learn discriminative features by using the outer product of two pooled features from different CNNs. Inspired by the bilinear pooling in [18], there appears an amount of works [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65] about representation learning with construction of high-order features. Gao et al. [37] proposed a compact bilinear pooling method to reduce feature dimensionality of high order representations, and more efficient compact bilinear pooling have been achieved by Shifted Random Macaullrin [62] and Kronecker Product [64]. Li et al. [55] proposed a matrix power normalized covariance method to explore effective high-order statistics, and further improved the training of global covariance pooling via iterative matrix square root normalization [56], which inspires a number of follow-upers [58], [61], [65]. Zheng et al. [59] proposed a deep bilinear transformation block to produce and aggregate intra-group high-order features based on semantic grouping of feature channels. Beyond exploiting intra-layer feature maps, Yu et al. [19] proposed a cross-layer bilinear pooling to enrich representations via fusing features from different granularities and spatial locations. Different from bilinear-based pooling, the methods in [38] and [39] both proposed a cross-granularity framework for learning a mixture of features from different levels, while Du et al. [66] proposed multi-granularity feature learning with a progressive training strategy.

2) Regularization-Based: Alternatively, regularization-based methods [46], [47], [48] adopt prevalent self-supervised learning as regularization for enhancing discriminate features learning. A number of metric learning based methods [20], [21], [42], [43], [44], [45] was proposed to regularize feature encoding with similarity constraints. Zhang et al. [42] adopted the squared Euclidean distance as similarity measurement between two $l_2$-normalized vectors and then generalized the triplet loss by constructing a triplet network. As traditional metric learning methods always adopting distance measure, Wang et al. [43] firstly proposed an angular constraint for deep metric learning by encoding the angular relation among triplets, which can be high-order relation and thus able to reveal more local structure. Methods in [44] and [21] both leveraged the multi-pair triplet loss to optimize the convergence, and Dubey et al. [45] proposed a pairwise confusion regularization by minimizing the squared Euclidean distance between the output distribution of random pairs for preventing over-fitting. Recently, Zhuang et al. [20] proposed an Attentive Pairwise Interaction architecture for learning the mutual feature between input pairs and further taking feature priorities into each account with a score ranking loss.

The proposed method falls into the latter group of regularization based feature learning. However, different from aforementioned approaches, our dynamic target relation graph as self-generated constraints reveals latent inter-class correlation based on class-level representations, rather than traditional instance-level similarity measure, which can alleviate sparse and long tailed data distribution and therefore improve robustness of visual representations.

III. DYNAMIC TARGET RELATION REGULARIZATION

Given a training set $\{(x_i, y_i)\}_{i=1}^{N}$, the problem of fine-grained image classification aims to learn a mapping function $\Phi : \mathcal{X} \rightarrow \mathcal{Y}$ that classifies any test instance into one of $K = |\mathcal{Y}|$ object categories (e.g., bird breeds in fine-grained bird classification [4]), with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ denoting any input image and its corresponding category label and N being the number of training instances. Convolutional neural networks based algorithms [1], [2], [3] have dominated the problem for one decade, which typically can be formulated into a cascade of one feature extractor $\Phi_{\text{fea}} : \mathcal{X} \rightarrow \mathbb{R}^d$ and a classifier $\Phi_{\text{cls}} : \mathbb{R}^d \rightarrow [0, 1]^K$ as

$$\Phi(x) = \Phi_{\text{cls}}(z) \circ \Phi_{\text{fea}}(x)$$

where $d$ denotes the dimension of the feature representation output $z \in \mathcal{Z}$ of $\Phi_{\text{fea}}(x)$. The feature vector can be mapped to a K-dimensional prediction vector $p = \Phi_{\text{cls}}(z) = [p_1, p_2, \ldots, p_K]$ by the classifier, subject to $\sum_{j=1}^{K} p_j = 1$.

Dependency in the label space has been verified its effectiveness on improving feature discrimination such as label relation graph [26], label smoothing [25], target codes [27]. However, existing methods adopted a static inter-class relation via professional prior knowledge or a soft labeling strategy,
which can be less flexible to reveal complex correlation across categories. As a result, this paper concerns on discovering latent correlation in the label space and thus proposes a novel dynamic target relation graph constraint on feature encoding $\Phi_{\text{fea}}(x)$, which accordingly update the interrelation graph between target classes compounded with feature learning.

The whole pipeline of deep representation learning regularized by dynamic target relation graphs is shown in Fig. 2, which can be divided into the backbone of feature encoding (typically ResNet-50 [2], ResNet-101 [2], DenseNet-161 [3] adopted in our experiments), generation of dynamic target relation graphs (see Sec. III-A), supervised representation learning using an auxiliary target relation aware constraint (see Sec. III-B). More specifically, generation of the dynamic graphs of inter-class relation can have the following two steps: 1) online generation of representation centers (illustrated in Fig. 2 (b)) and 2) graph construction on the anchored centers (illustrated in Fig. 2 (c)). Beyond the popular class supervision using the cross entropy (CE) loss $L_{\text{CE}}$ as follows:

$$L_{\text{CE}} = - \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} y_{i,j} \log \frac{\exp(p_{i,j})}{\sum_{j'} \exp(p_{i,j'})},$$

(2)

the aforementioned two modules are additionally supervised to minimize the intra-class variation and model inter-class relation respectively in our scheme, which thus encourage discriminative features.

During testing, an unseen image is fed into a cascade of feature encoder $\Phi_{\text{fea}}(x)$ and a classifier $\Phi_{\text{cls}}$ to predict its class probability directly.

A. Construction of Dynamic Target Relation Graphs

1) Generation of Class-Level Feature Centers: Before discovering the relation between target classes, the class-level representation (i.e., class center in feature space) plays an important role, as it is sensitive to data distribution. In [28], a learnable scheme of adaptive generation of class-wise feature centers is introduced to formulate the generation into parametric estimation. Such a method can be effective to minimize intra-class variations, but optimization on those center parameters suffers from the challenge of sparse and imbalanced data distribution in practice. The aforementioned observation encourages us to adopt the mean feature vector of samples in each class as the class-level representation, which is robust and easy to obtain during the iteration of parameters updating.

Technically, we first define the class feature center matrix $C^t \in \mathbb{R}^{K \times d}$ for $K$ classes, where the representation of each class is a $d$-dimensional vector and the superscript $t$ denotes the $t$-th epoch during model training. The center matrix $C^0$ can be typically initialized for all classes with random Gaussian parameters, which indicates that the centers for all classes are random points in the feature space. The class-level center matrix $C^t$ is iteratively updated by averaging all the features of samples from the same class during every epoch. The center matrix $C$ can be employed to define a non-parametric online center loss (OCL):

$$L_{\text{OCL}} = \frac{1}{N} \sum_{i=1}^{N} \| z_i - C^{t-1}_{y_i} \|_2^2,$$

(3)

where $z_i$ is the feature vector of any sample and $C^{t-1}_{y_i}$ is its corresponding class-level representation vector from the center matrix $C$ generated in the last epoch (i.e., $(t-1)$-th epoch). $\| \cdot \|_2$ denotes the Euclidean norm. More details are given in Algorithm 1. Note that, our generation of class-level representations is similar to prototype learning proposed in [67]. However, our method is designed in a non-parametric manner, whereas prototype learning usually requires optimization of

![Fig. 2. Overview of our proposed Dynamic Target Relation Graphs (DTRG) method. (a) A general backbone for feature extraction and supervised learning, and more details are illustrated in the CNN framework on the right handside. (b) Our module to generate Online Class Feature Centers, where $z_i, i = 1, 2, \ldots, 6$ is the sample feature vector and $c_t$ denotes the corresponding class feature center. (c) The proposed module to construct inter-class target relation graph, where $c^{t-1}$ is the class-level feature from the last epoch (i.e., the $t-1$-th epoch), $z^t_i$ is the sample feature from current epoch (i.e., the $t$-th epoch), the target relation graph $G^t$, and graph-based representation $S^t$. (d) Our Online Center Loss for reducing the intra-class distance.](image-url)
Similarity function as follows

class-level feature centers, we firstly consider the Cosine
which can thus be robustness against outliers.
fifying with relative positioning of class-level representations,
encoding towards awareness of sample distribution via simpli-
feature space. Such a setting constrains optimization of feature
deep representation learning with underlying manifolds in the
considered as a self-supervised supervision for regularization of
relative distance of class-level representations, which is con-
categories. Given the feature centers for all categories at each
visual task, there always exist some extremely similar cat-

generated from last epoch (the

centers of all categories. Specifically, the feature
relative distance between these samples’ features and feature
encoder

more learnable parameters and thus can be more sensitive to
data distributions.

2) Construction of Inter-Class Relation Graphs in Target Space: Since different categories are always of varying
degrees of relation correlation, especially in our fine-grained
visual task, exist some extremely similar categories. Given the feature centers for all categories at each
epoch, we can construct a target relation graph by measuring
relative distance of class-level representations, which is con-
sidered as a self-supervised supervision for regularization of
deep representation learning with underlying manifolds in the
feature space. Such a setting constrains optimization of feature
encoding towards awareness of sample distribution via simpli-
fying with relative positioning of class-level representations,
which can thus be robustness against outliers.

To measure target relation with relative distance of
class-level feature centers, we firstly consider the Cosine
Similarity function as follows:

\[
\cos(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\|_2 \|z_j\|_2}.
\]  

(4)

To keep the monotonicity and non-negativity of similarity metric, we further define our similarity function as follows:

\[
s(z_i, z_j) = e^{\frac{z_i \cdot z_j}{\tau}},
\]  

(5)

where \(\hat{z}_i = \frac{z_i}{\|z_i\|_2}\) is the normalized feature of \(z_i\) and \(\tau\) is a
temperatures parameter for controlling the variance. It is noted
that other distance metrics such as squared Euclidean distance
and Gaussian kernel function can be employed, but the cosine
similarity can perform better to discover a similar distribution
of feature dimensions. As a result, we employ Eq. (5) in our
scheme.

In mathematics, given a feature center matrix \(C^{t-1}\) for
updating features in the \(t\)-th epoch, the \(j\)-th row \(C_j \in \mathbb{R}^d\)
of \(C\) being the mean feature center within the corresponding
category, where \(j = 1, 2, \ldots, K\) and \(t = 1, 2, \ldots, T\), the target
relation graph matrix \(G^t \in \mathbb{R}^{K \times K}\) can thus be constructed as the following:

\[
G^t_{kl} = s(C^t_{l1}, C^t_{l-1}) = e^{\frac{C^t_{l1} \cdot C^t_{l-1}}{\tau}},
\]  

(6)

where \(G^t_{kl}\) denotes the similarity connected between any two
iteration-varying centers \(C^t_{l1}\) and \(C^t_{l-1}\), as we obtain the
feature centers using features at the \(t-1\)-th epoch; and \(G^t\) is a
dynamic target relation graph.

Having the dynamic self-generated regularization objective
(i.e., \(G\)) at each epoch, the samples \(x\) are first fed into the deep
feature encoder \(\Phi_{fe}(x)\) to generate their features \(z\), which are then transformed into graph representations by computing
relative distance between these samples’ features and feature
centers of all categories. Specifically, the feature \(z^t_i = \Phi_{fe}(x^t_i)\)
for sample \(x^t_i\) at the \(t\)-the epoch, we can obtain the graph
representation of the sample \(z^t_i\) between the sample feature
\(z_i\) and any centers \(C^t_{l1}\) generated from last epoch (the \(t-1\)-th
epoch).

\[
S^t_{ij}(z^t_i) = s(z^t_i, C^t_{l-1}) = e^{\frac{z^t_i \cdot C^t_{l-1}}{\tau}},
\]  

(7)

where \(i = 1, 2, \ldots, N\), \(j = 1, 2, \ldots, K\) and \(t = 1, 2, \ldots, T\). In this way, the graph representation
\(S^t_{ij} = [S^t_{1j}(z^t_1), S^t_{2j}(z^t_2), \ldots, S^t_{Kj}(z^t_K)] \in \mathbb{R}^K\) can be
generated from the sample’s feature \(z^t_i\) and the feature center
matrix \(C^{t-1}\).

B. Feature Regularization With Target Relation Graphs

In the previous section, the target relation graph \(G^t\) and
the graph-based representation \(S^t_{ij}\) from \(z_i\) of sample \(x_i\) at the
\(t\)-th epoch are generated respectively. We consider maximizing
these similarities of class-relation graph by minimizing the
distance between target relation graph \(G^t_{yi}\) and inter-class
graph representation \(S^t_{ij}\) of \(x_i\), where \(G^t_{yi} \in \mathbb{R}^K\) is the
vector to model target correlation anchored on label \(y_i\) of
sample \(x_i\).

To measure distance between graphs, we first choose the
popular squared Euclidean distance based on the theory analysis
from [45]. We can formulate the similarity constraint loss function of inter-class relation graph – graph similarity
loss (GSL) as follows:

\[
L_{\text{GSL}} = \frac{1}{N} \sum_{i=1}^{N} \|S^t_{ij} - G^t_{yi}\|^2_2.
\]  

(8)
Alternatively, the Kullback-Leibler (KL) Divergence is adopted as similarity metric as:

\[
D_{\text{KL}}(P\|Q) = \sum_{x \in \mathcal{X}} P(x) \ln \frac{P(x)}{Q(x)},
\]

where \(P\) and \(Q\) are probability distributions from the same space \(\mathcal{X}\). Before using the KL Divergence to encode target relation into imagery features, both relation graph vectors \(S'_{ij}\) and \(G'_{ij}\) are normalized by their norms being 1. As a result, the normalized relation graphs are depicted as:

\[
\hat{S}'_{ij} = \frac{S'_{ij}}{\sum_j S'_{ij} e^{\frac{-e^{\gamma_{ij}}}{\tau}}},
\]

\[
\hat{G}'_{ij} = \frac{G'_{ij}}{\sum_j G'_{ij} e^{\frac{-e^{\gamma_{ij}}}{\tau}}},
\]

In this way, the KL divergence between \(G'_{iy}\) and sample graph \(\hat{S}'_{i}\) can be formulated as the following:

\[
D_{\text{KL}}(\hat{G}'_{iy}||\hat{S}'_{i}) = \sum_{j=1}^{K} \hat{G}'_{ij} \ln \frac{\hat{G}'_{ij}}{\hat{S}'_{ij}},
\]

where \(\hat{S}'_{i}\) denotes the \(i\)-th normalized sample graph and \(\hat{G}'_{iy}\) is its corresponding target graph. The graph similarity loss function based on the KL Divergence can thus be written as:

\[
L_{\text{GSL}} = \frac{1}{N} \sum_{i=1}^{N} D_{\text{KL}}(\hat{G}'_{iy}||\hat{S}'_{i}).
\]

Both of the Square Euclidean distance and the KL-divergence can be used to constrain the similarity of inter-class relation graph, which are compared in our experiments (see Sec. IV-D) for evaluation on the better choice of graph similarity loss.

Beyond the conventional one-hot classification supervision signal, with the loss \(L_{\text{GSL}}\) as an extra self-generated objective, the deep representation learning can be regularized to incorporate target relation. It is worth mentioning here that, our graph similarity constraint can also be regarded as a kind of self-distillation method via self-supervised target relation regularization, which is different from traditional graph distillation [68], [69], [70] that always requires a larger model for teacher graph learning to guide a student network.

C. Inter-Class Relation Augmentation

To further enrich the inter-class relation, we introduce an inter-class relation augmentation. Our inter-class relation augmentation is inspired by recent success of the classical data augmentation Mixup [29]. Specifically, the Mixup arguments more training images by linearly combining two random input images \(x_a, x_b\) from different classes and their corresponding labels \(y_a, y_b\), which can be described as:

\[
\tilde{x}_i = (1 - \lambda)x_a + \lambda x_b, \quad \tilde{y}_i = (1 - \lambda)y_a + \lambda y_b,
\]

where \(\lambda \sim \text{Beta}(\alpha, \alpha)\), for \(\alpha \in (0, \infty)\). Note that, a number of mixup-style data augmentation methods such as CutMix [30] and SnapMix [22] can also be depicted by Eq. (14), while they differs from the way to generate \(\lambda\). As a result, these mixup augmentation strategies can be adopted in our inter-class relation augmentation, whose performance differences are compared in our experiments (See Table XI).

We follow the same data augmentation processing as Mixup [29] to generate training samples \(\{\tilde{x}_i, \tilde{y}_i\}^N\), and then we leverage soft labels of these mixed images for creating more diverse target relation graph. Firstly, we obtained the feature \(\tilde{z}_i\) of augmented image \(\tilde{x}_i\).

\[
\tilde{z}_i = \Phi_{\text{fea}}(\tilde{x}_i).
\]

For features \(\tilde{z}\) from mixed images, we use the pipeline shown in Algorithm 2 to generate dynamic class centers matrix \(C'\). Then, we can calculate a mixed version of non-parametric online center loss as:

\[
L_{\text{OCL}} = \frac{1}{N} \sum_{i=1}^{N} (1 - \lambda)\|\tilde{z}_i - C'_{ya}\|_2^2 + \lambda \|\tilde{z}_i - C'_{yb}\|_2^2.
\]

Beyond Eq. (16), other mixed options for \(L_{\text{OCL}}\) can be employed such as \(\frac{1}{N} \sum_{i=1}^{N} \|\tilde{z}_i - (1 - \lambda)C'_{ya} - \lambda C'_{yb}\|_2^2\). Here, we select Eq. (16) in our scheme in view of its simple form to make representation learning easy.

For generating mixed sample inter-class relation graph \(S'_{iy}(\tilde{z}_i)\) and its corresponding target relation graph \(G'_{iy}\), the following equations are utilized:

\[
S'_{iy}\left(\tilde{z}_i\right) = s\left(\tilde{z}_i, C'_{iy}^{-1}\right),
\]

\[
G'_{ya, j} = s\left(C'_{ya}^{-1}, C'_{y}^{-1}\right),
\]

\[
G'_{yb, j} = s\left(C'_{yb}^{-1}, C'_{y}^{-1}\right),
\]

where \(C'_{iy}^{-1} = C'_{y}^{-1}[*], \ j \in [1, 2, \ldots, K]\), and * represents the class index.

Similarly, we can construct a mixed graph similarity loss based on the former relation graphs in Eq. (17) and Eq. (18).

Therefore, we use the Square Euclidean distance to formulate the following mixed similarity loss function as:

\[
L_{\text{GSL}} = \frac{1}{N} \sum_{i=1}^{N} \|S'_{iy} - G'_{ya, j}\|_2^2 + \lambda \|S'_{iy} - G'_{yb, j}\|_2^2.
\]

Note that, we can also construct the KL-divergence based mixed similarity loss function, but superior performance of the Square Euclidean distance to than KL-divergence is achieved in our experiments (see Fig. 6 for more details). Owing to such an adaptation of mixup style data augmentation into the proposed dynamic target relation graph scheme, a diverse data to model complex target relation can further boost performance of fine-grained classification, which is verified in our experiments.

D. Model Training

The total loss for our method can be formulated as:

\[
L = L_{\text{CE}} + \beta L_{\text{OCL}} + \eta L_{\text{GSL}}.
\]

where \(\beta\) and \(\eta\) are trade-off hyper-parameters. As all terms of the object function in Eq. (20) are differentiable regardless...
Algorithm 2 Model Training Processing With Inter-Class Relation Augmentation

**Input:** Dataset $D_{\text{train}} = \{(x_i, y_i)\}$, the feature extractor $\Phi_{\text{fea}}$, the classifier $\Phi_{\text{cls}}$, training epochs $T$, class number $K$, feature dimension $d$, start epoch $T_{th}$, hyper-parameter $\alpha$ for mixup style data augmentation

**Initialize:** Class feature center matrix $C^0 \in \mathbb{R}^{K \times d}$ with random Gaussian parameters

1: for current epoch $t = 1$ to $T$
2: Initialize the middle matrix $M \in \mathbb{R}^{K \times d}$ and the count matrix $V \in \mathbb{R}^{K \times 1}$ both with zeros.
3: for iteration $= 1$ to $T$
4: Random sample a batch training data $B \subset D_{\text{train}}$
5: for $i = 1$ to $|B|$ do
6: Generate mixed sample $\{\tilde{x}_i, \tilde{y}_i\}$ by Eq.(14)
7: Obtain the predicted feature $\tilde{z}_i$ by Eq.(15)
8: Update $M[y_i] \leftarrow M[y_i] + (1-\lambda)\tilde{z}_i$ and $M[y_b] \leftarrow M[y_b] + \lambda \tilde{z}_i$
9: Update $V[y_a] \leftarrow V[y_a] + (1-\lambda)$ and $V[y_b] \leftarrow V[y_b] + \lambda$
10: Compute the center loss $L_{\text{CCL}}$ by Eq.(16)
11: Compute the similarity loss $L_{\text{CCL}}$ by Eq.(19)
12: Obtain the prediction vector $\tilde{p}_i = \Phi_{\text{cls}}(\tilde{z}_i)$
13: end for
14: if $t < T_{th}$ then
15: Compute total loss according by Eq.(2)
16: else
17: Compute total loss according by Eq.(20)
18: end if
19: Backward to update the parameters of model
20: end for
21: for $j = 1$ to $K$
22: Update $C^t[j] \leftarrow M[j]/V[j]$
23: end for
24: end for
25: return The learned parameters of $\Phi_{\text{fea}}$ and $\Phi_{\text{cls}}$

of selection of two distance metric options for the $L_{\text{GSL}}$ and whether to use inter-class correlation augmentation. Consequently, the model can be trained in an end-to-end learning manner with stochastic gradient descent algorithm [71]. To learn a relatively stable centers matrix $C$, we can use a warm-up strategy by training the model only with $L_{\text{CE}}$ loss function for the first $T$-th epochs, which can make sure that the center matrix to avoid making excessive deviation. The whole training processing with inter-class relation augmentation is summarized in Algorithm 2.

### IV. EXPERIMENTS

In this section, to evaluate the effectiveness and generalization of our proposed method, we conduct experiments on diverse public benchmarks of multiple visual classification tasks. We firstly investigate details of datasets and experimental settings. Secondly, the proposed method and the state-of-the-art methods are evaluated and compared on popular benchmarks. Thirdly, we verify superior robustness of our method for sparse and imbalanced data. Moreover, ablation studies for the proposed components are evaluated and compared, to verify their effectiveness. Finally, visualizations based on t-SNE [72] and Grad-CAM [73] are presented for better understanding our motivation.

#### A. Datasets and Experimental Setting

In the problem of fine-grained image classification, we evaluate our method on the following widely used datasets: CUB-200-2011 (CUB) [4], Stanford Cars (CAR) [6], FGVC Aircraft (AIR) [7], and large-scale long-tailed iNaturalist 2018 [74]. In our experiments, we follow the data splits provided in [22], whose details are provided in Table I, while the example fine-grained images are visualized in Fig. 3. More experiments are conducted on the problem of generic image and texture classification, on two generic image classification benchmarks – the CIFAR [75] and the ImageNet [76] and also two texture image datasets – the DTD [77] and the MIT Scene (MIT) [78]. The dataset statistics are summarized in Table I, and training/testing splits for the DTD and the MIT datasets from their official websites are adopted in our experiments. Only category labels of training data are adopted as supervision signals without using any additional prior information.

1) Data Pre-Processing: The raw images from the CUB are firstly resized to $512 \times M$ without warping, where $M$ is the length of a longer side, while images from the CAR, the AIR, the DTD and the MIT datasets are resized to $512 \times 512$ directly with warping, following the same setting in [22] and [40]. These images are then cropped with size $448 \times 448$ for batch training and testing. For the ImageNet and the iNaturalist 2018 datasets, the images are cropped to size $224 \times 224$. During training, we adopt random cropping and random horizontal flipping for data augmentation by following most of previous works [22], [40], [79]; while we only adopt center cropping during testing as [22]. For the CIFAR datasets,
TABLE I
THE STATISTICS OF DATASETS

| Dataset         | # classes | # training data | # testing data |
|-----------------|-----------|-----------------|----------------|
| CUB [4]         | 200       | 5,994           | 5,794          |
| CAR [6]         | 196       | 8,144           | 8,041          |
| AIR [7]         | 100       | 6,667           | 3,333          |
| iNaturalist 2018 [74] | 8142      | 437,513         | 24,426         |
| CIFAR10/100 [75] | 10/100    | 50,000          | 10,000         |
| ImageNet [76]   | 1000      | 1,281,167       | 50,000         |
| DTD [77]        | 47        | 1.880           | 1.340          |
| MIT [78]        | 67        | 5,360           | 1,340          |

TABLE II
COMPARISON WITH STATE-OF-THE-ART METHODS ON CLASSIFICATION ACCURACY (%) WITH CUB, CAR AND AIR BENCHMARKS. THE REPORTED RESULTS OF OUR METHOD ARE OBTAINED BY COMBINING OUR PROPOSED DYNAMIC TARGET RELATION GRAPH (DTRG) WITH BASELINE+MID AND MIXUP-BASED DATA AUGMENTATION. THE BEST RESULTS FOR EACH DATASET ARE MARKED IN BOLD.

| Method       | Base Model | Year  | CUB (%) | CAR (%) | AIR (%) |
|--------------|------------|-------|---------|---------|---------|
| BiLinear CNN [18] | VGG16      | 2015  | 84.1    | 91.3    | 84.1    |
| MoNet [56]    | VGG16      | 2018  | 86.4    | 91.8    | 89.3    |
| HBP [19]      | VGG16      | 2018  | 87.1    | 93.7    | 90.0    |
| MGD [38]      | VGG19      | 2015  | 81.7    | -       | 82.5    |
| RA-CNN [17]   | VGG19      | 2017  | 85.3    | 92.5    | -       |
| MA-CNN [34]   | VGG19      | 2017  | 86.5    | 92.8    | 89.9    |
| sSQRT-COV [56] | ResNet50   | 2018  | 88.1    | 92.8    | 90.0    |
| DFL [80]      | ResNet50   | 2018  | 87.4    | 93.1    | 91.7    |
| DCL [46]      | ResNet50   | 2019  | 87.8    | 94.3    | 93.0    |
| LIO [47]      | ResNet50   | 2019  | 88.0    | 94.5    | 92.7    |
| S3N [35]      | ResNet50   | 2019  | 88.5    | 94.7    | 92.8    |
| MGE-CNN [39]  | ResNet50   | 2019  | 88.5    | 95.3    | 93.9    |
| MC-Loss [40]  | ResNet50   | 2020  | 87.3    | 93.7    | 92.6    |
| PMG [66]      | ResNet50   | 2020  | 89.6    | 95.1    | 93.4    |
| GaRD [81]     | ResNet50   | 2021  | 89.6    | 95.1    | 94.3    |
| SVD-Trunc [61] | ResNet50   | 2021  | 87.3    | 93.0    | -       |
| SPS [82]      | ResNet50   | 2021  | 88.7    | 94.9    | 92.7    |
| DTRG (Ours)   | ResNet50   | -     | 88.8    | 95.2    | 94.1    |
| sSQRT-COV [56] | ResNet101  | 2018  | 88.7    | 93.3    | 91.4    |
| DBTNet [59]   | ResNet101  | 2019  | 88.1    | 94.5    | 91.6    |
| CIN [83]      | ResNet101  | 2020  | 88.1    | 94.5    | 92.8    |
| SnapMix [22]  | ResNet101  | 2021  | 89.5    | 94.8    | 94.1    |
| DTRG (Ours)   | ResNet101  | -     | 89.7    | 95.4    | 94.2    |
| PC [45]       | DenseNet161| 2018  | 86.9    | 92.9    | 89.2    |
| FDL [84]      | DenseNet161| 2020  | 89.1    | 94.0    | 91.3    |
| API [100]     | DenseNet161| 2020  | 90.0    | 95.3    | 93.9    |
| DTRG (Ours)   | DenseNet161| -     | 90.1    | 95.7    | 94.8    |

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY (%) ON THE iNATURALIST 2018 DATASET USING THE ResNet50 BACKBONE

| Method  | Schedule | iNaturalist 2018 |
|---------|----------|------------------|
|         |          | top-1 (%) | top-5 (%) |
| Focal [87] | SGD      | 58.03     | 78.65   |
| CB-Focal [85] | SGD  | 61.12     | 81.03   |
| LDAM [86] | SGD      | 64.58     | 83.52   |
| IB [79]   | SGD      | 65.39     | 84.98   |
| Baseline  | SGD      | 64.21     | 84.35   |
| DTRG (Ours) | SGD  | 65.54     | 85.99   |
| LDAM [86] | DRW      | 68.00     | 83.18   |
| Baseline  | DRW      | 67.03     | 84.75   |
| DTRG (Ours) | DRW  | 69.47     | 87.36   |

we randomly crop the images into size 32 × 32 with 4 pixels padding at each side and with random horizontal flipping by following the basic data augmentation operations used in [2] and [25].

2) Backbone and Baseline: For fine-grained image classification on the CUB, the CAR and the AIR, to better evaluate the proposed DTRG method, three popular networks are employed as backbone, i.e., ResNet50 [2], ResNet101 [2] and DenseNet161 [3], which are all pre-trained on the ImageNet [76]. Beyond the backbone networks with only one simple linear classifier as a vanilla model, inspired by several recent works [22], [39] to boost the performance of recognition, a double-branch strategy is adopted using the CE loss to supervise features from different network depth, as our advanced baseline, which is termed as backbone+mid. In our experiments, we follow the implementation from Sanpmix [22] by adding an extra branch with another linear classifier on features of the mid-level layer of backbone networks. Specifically, such a mid-level branch includes a ConvBlock with one Conv 1 × 1 layer and one ReLU module, a Max Pooling layer and a Linear Classifier with one layer. For the ResNet, this mid-level branch is placed after the 4-th block of backbone network, while for the DenseNet, that is placed after the penultimate denseblock of backbone network. By incorporating the new branch in the mid-level level of backbone networks, the double-branch network architecture can capture cross-granularity features. Note that, the gradients from the mid-level branch are blocked to the main backbone during training. In inference, we use the ensemble of prediction outputs from two classifier to generate final predictions. On the iNaturalist 2018, our proposed DTRG is compared with baseline methods using ResNet50 [2] as their backbone, as well as several state-of-the-art methods [79], [85], [86]. For the CIFAR datasets, the baselines are based on the backbone of ResNet18/34/50/101 [2] respectively, while on the ImageNet dataset, our proposed DTRG is compared with the baselines using ResNet50/101 backbone networks.

For texture classification on the DTD and the MIT datasets, we compare our method with the baseline based on the pre-trained ResNet50 [2]. Note that our models for the CIFAR, the ImageNet and the iNaturalist datasets are all trained from scratch.

3) Training Details: For computational efficiency, in all our fine-grained and texture recognition experiments, the models without using mixup-based target relation augmentation are trained for 100 epochs, while only these with mixup-based augmentation are trained for 200 epochs to exploit more diverse target relation for superior performance. The hyperparameter $\alpha$ of Eq. (14) is set as 0.1 for Mixup and 1 for CutMix and SnapMix, respectively. The parameter $r$ in similarity constraint (Eq. (6) and Eq. (7)) is set as 1 for the vanilla backbone and 2 for backbone+mid; while the parameter $\beta$ of Eq. (20) for controlling the influence of online center loss...
is set as $1e - 3$ by our ablation studies; the parameter $\eta$ of Eq. (20) is set as 1 for squared Euclidean Distance by our ablation experiments. $T_{th}$ in Algorithms 1 and 2 is both set as 2 for better iterating the class feature center matrix during the first epoch, verified by our ablation studies (see Table IX). During training, we adopt stochastic gradient descent (SGD) for network operation with momentum of 0.9 and weight decay of $1e - 4$. The batch size in all our experiments is set as 16. The initial learning rate is set as 0.01 for new parameters and 0.001 for pre-trained parameters, which descend following the multi-step schedule with decay factor as 0.1. The milestones for decaying learning rate are set as [40, 70] for 100 epochs and [80, 150, 180] for 200 epochs. In the experiments on the CIFAR, we use the SGD optimizer with momentum 0.9 and weight decay $5e - 4$ to train all the models for 200 epochs with a batch size of 128. The learning rate is initially set as 0.1 and decays at the 100th and 150th epoch by a factor of 0.1, respectively. The other hyper-parameters in these experiments are set as same as introduced before. In the experiments on the ImageNet and the iNaturalist 2018, we train all the models with batch size 256, initial learning rate 0.1, and cosine learning rate schedule for better optimization. The training epochs for the ImageNet dataset is set as 100, while 200 for the iNaturalist dataset. Due to the large scale size of both two datasets, we set the hyper-parameters $T_{th}$ to be half of the size of total epochs for better generating feature centers. The other hyper-parameters $\tau$ is set as 2, $\beta$ is set as $2e - 4$, and $\eta$ is set as $2e - 3$ for balancing the loss value of our DTRG for the iNaturalist dataset, while $\eta$ is set as 0.1 on the ImageNet dataset.

**B. Comparative Evaluation**

1) **Evaluation on Fine-Grained Classification:** Comparative evaluation on the CUB-200-2011, the Stanford Cars, and the FGVC-Aircraft datasets is reported in Table II. Evidently, our method can consistently achieve superior performance to the state-of-the-art methods on the three datasets, i.e., 90.1%, 95.7% and 94.8% on the CUB, the CAR, and the AIR datasets respectively. Moreover, with three popular backbone networks (the ResNet50, ResNet101, and DenseNet161), the proposed DTRG method can always perform better than existing methods, which further verifies its effectiveness. Furthermore, to verify the effectiveness of our method on large-scale fine-grained benchmark, we conduct one more experiment on the iNaturalist 2018 dataset. Comparative evaluation with the state-of-the-art methods is reported in Table III, where our DTRG method can always outperform the baseline and also be superior to existing works, especially when training with the training schedule DRW introduced in [86]. The results on the iNaturalist again demonstrate our motivation.

2) **Evaluation on Ordinary Image Classification:** As our method is not limited to fine-grained classification, we evaluate the proposed DTRG on the CIFAR10/100 and the ImageNet of general image classification problem with different backbone architectures, respectively. Comparative evaluation of different ResNet variants with or without our DTRG method on the CIFAR10/100 is reported in Table IV, where our DTRG can consistently beat the baseline methods. Similar results on the large-scale ImageNet in Table V can be observed with ResNet50/101 as the backbone. The results on all datasets can confirm that our DTRG is generic and can be inserted in existing deep networks for different visual classification tasks.

3) **Evaluation on Texture Classification:** Comparative evaluation on the DTD and the MIT datasets is reported in Table VI. Superior performance of our DTRG with the ResNet50 backbone can be achieved to the baseline and the state-of-the-art methods, i.e., TKPF [64] and SPS [82]. On the DTD, we report the mean accuracy across the 10 splits provided in [77], and our methods can outperform the baseline by about 3%. The results on the two texture datasets demonstrate that our methods can improve the representation learning on diverse visual classification tasks, specifically fine-grained scene recognition and texture classification.

4) **Comparison With Label Smooth and Center Loss:** As label smooth and center loss algorithms are our direct competitors, for a fair comparison of our DTRG method with the vanilla label smoothing [24], online label smooth
TABLE VII

| Method                  | CUB (%) | CAR (%) | AIR (%) |
|-------------------------|---------|---------|---------|
| Label Smooth [24]       | 86.0    | 93.4    | 92.0    |
| OLS [25]                | 86.3    | 93.3    | 91.2    |
| Center Loss [28]        | 86.4    | 93.5    | 91.4    |
| DTRG (Ours)             | **87.3**| **94.3**| **92.0**|

Fig. 4. Classification performance under varying sample size of each class.

(OLS) [25] and Center Loss [28], we conduct an experiment on the CUB-200-2011, the Stanford Cars, and the FGVC-Aircraft datasets with the ResNet50 backbone under the same settings. Results are illustrated in Table VII. It is observed that our DTRG method can achieve the best performance among all comparative methods. These results can confirm again the superiority of our DTRG method to these related works. Moreover, the better effectiveness of our DTRG method can be gained because of non-parametric center generation, which can be more stable than Center Loss [28] trained with additional learnable parameters.

C. Robust Against Sparse and Imbalanced Data

In order to verify robustness of our method against sparse and imbalanced data distribution, we conduct two more experiments on the CUB-200-2011 dataset with the ResNet50 backbone. As Table I shown, the CUB-200-2011 dataset contains total 200 classes, where each class has a relative balanced distribution (about 30 training samples for each class), which thus requires to adapt the CUB dataset to a more challenging sparse and imbalanced setting.

For evaluation on sparse data, we set the sparsity of the training set with a limited size of samples for each class (removing 50% to 90% of training samples), whereas the whole validation set is adopted in our experiments. As shown in Fig. 4, our DTRG method shows superior robustness against sparse data, which is credited to utilize structural correlation in the target space, compared to its baseline competitor. Specifically, as mentioned in the introduction, representation learning for sparse classes can be supported by samples of other correlated semantic classes, as information across classes can be shared via latent target relation graph regularization.

D. Ablation Studies

In the total loss function in Eq. (20), there are three loss terms – the CE loss in Eq. (2) for typical supervision on classification, the Online Center loss in Eq. (3) for minimizing the intra-class feature variations, and the Graph Similarity loss in Eq. (8) for incorporating inter-class relations. In this section, we first compare the vanilla baseline and the advanced baseline with an extra mid-level branch. The effects of modules in our proposed method with different hyper parameters are investigated and evaluated. Finally, the inter-class relation augmentation is compared on three backbone networks to demonstrate its effectiveness.

1) Impact of the Proposed Components: We conduct ablation studies to demonstrate the effectiveness of the two key components in our model, including the non-parametric online center loss and the graph similarity constraint. Ablation studies are conducted on six different baselines, which adopt three different backbone networks, ResNet50, ResNet101 and DenseNet161 with or without using mid-level branch on three different fine-grained datasets. From Table VIII, we can find...
out that the effectiveness of the proposed modules with different backbones on multiple datasets are verified. Especially, our models based on backbone+mid can always get the better performance than those based on vanilla backbone, which indicates that additionally combining features from mid-level can further improve discrimination of features. What’s more, the deeper the network structure, the better performance. With the DenseNet161+mid backbone, we can achieve the state-of-the-art performance on all three datasets. As depicted in Eq. (20), two hyper parameters $\beta$ and $\eta$ for non-parametric online center loss and graph similarity loss need to be tuned respectively. Note that we have proposed two options of graph similarity constraint, and hence we replace the hyper parameter $\eta$ with $\eta_{eu}$ and $\eta_{kl}$ for the two graph similarity constraints, corresponding to the squared Euclidean distance and the KL Divergence.

a) Effects of hyper parameter $\beta$: In the first experiment, we add our non-parametric online center loss to the baseline as the only regularization, and vary $\beta$ from $1e-4$ to $1e-2$. The experiment is conducted on the CUB-200-2011 dataset with the ResNet50 backbone. The results shown in Fig. 6 (a), where the performance can generally be improved by adding our non-parametric online center loss. We can also observe that when $\beta = 1e - 3$, the best result is gained, which can outperform the baseline by 1.01% on classification accuracy. It is noted that $\beta = 1e - 3$ is fixed for other experiments.

b) Effects of hyper parameter $\eta$: To explore the influence of different similarity constraints, we set up two groups of comparative experiments for $\eta_{eu}$ and $\eta_{kl}$, respectively. Considering different properties, the parameter scales of $\eta_{eu}$ and $\eta_{kl}$ are different. That is, $\eta_{eu}$ can be varied from 0.1 to 1.2 for similarity constraint based on the squared Euclidean distance, and $\eta_{kl}$ is set from 1 to 25 with an approximately equal interval for similarity constraint method based KL Divergence. Experiments are conducted by adding our graph similarity loss as the only regularization with the above settings of parameter. The results are visualized in Fig. 6(b) and (c), which can be evidently observed that both similarity constraints can outperform baseline consistently. Moreover, it is concluded that graph similarity constraint based on squared Euclidean distance can perform slightly better than that based on KL Divergence. The best result is achieved when we adopt the squared Euclidean distance with $\eta_{eu} = 1$, which outperforms the baseline by 1.55%. As a result, the squared Euclidean distance is employed in our graph similarity constraint with $\eta = 1$ in other experiments.

c) Effects of starting epoch $T_{th}$: To explore the effects of different start epoch on classification performance when using our DTRG method, we set up a group of experiments with varying $T_{th}$ from 2 to 50 on the CUB-200-2011 dataset. Table IX show that the best results can be achieved when the start epoch $T_{th}$ is set to be a small value (e.g., 2 and 3), which can be credited to good yet diverse initial feature centers of target relation graph by the pre-trained models. Positive effects of feature regularization can be reduced with $T_{th}$ increasing, due to lack of exploration of latent target relation with convergence of feature centers. However, with any $T_{th}$, our DTRG can consistently outperform its baseline, which can again verify the superiority of our method.

2) Effects of Error Gradients From Extra Supervision on Middle Layers: Inspired by existing works [22], we add

| Backbone        | OCL | GSL | CUB (%) | CAR (%) | AIR (%) |
|-----------------|-----|-----|---------|---------|---------|
| ResNet50        |     | ✓   | 85.6    | 92.9    | 90.7    |
|                 | ✓   | ✓   | 87.3    | 94.3    | 92.0    |
| ResNet50+mid    | ✓   | ✓   | 87.7    | 93.6    | 91.5    |
|                 | ✓   | ✓   | 87.9    | 94.1    | 92.3    |
|                 | ✓   | ✓   | 88.3    | 94.8    | 93.0    |
| ResNet101       | ✓   | ✓   | 86.4    | 93.2    | 91.4    |
|                 | ✓   | ✓   | 87.1    | 93.8    | 92.1    |
|                 | ✓   | ✓   | 87.7    | 94.4    | 92.8    |
| ResNet101+mid   | ✓   | ✓   | 87.9    | 94.0    | 92.2    |
|                 | ✓   | ✓   | 88.1    | 94.2    | 92.6    |
|                 | ✓   | ✓   | 88.6    | 94.9    | 93.1    |
| DenseNet161     |     | ✓   | 87.3    | 93.3    | 92.4    |
|                 |     | ✓   | 88.2    | 94.0    | 92.9    |
|                 | ✓   | ✓   | 88.4    | 94.8    | 93.5    |
| DenseNet161+mid | ✓   | ✓   | 88.4    | 93.6    | 92.7    |
|                 | ✓   | ✓   | 89.0    | 94.2    | 93.1    |
|                 | ✓   | ✓   | 89.0    | 94.8    | 94.0    |

Fig. 6. Effects of hyper parameters for non-parametric online center loss and graph similarity loss with squared Euclidean distance or KL divergence.
another classification branch in the middle of our backbone network to boost classification performance. Since generality of the proposed DTRG scheme, our method can also be applied to regularize mid-level feature encoding with self-supervised structural relation in target space. We choose the ResNet50+mid as our backbone and conduct experiments on the CUB-200-2011 dataset, whose results are shown in Table X. Ablation studies on the online class loss and the geometric similarity loss are shown in the Table X, where w/o detach and w/ detach indicate whether gradients of errors generated from supervision on the mid-level layers will be propagated backward to the backbone networks or not. In other words, the models w/ detach can be considered as an ensemble of classifiers on feature encoding with different layers, while the models w/o detach use extra supervision on mid-level feature encoding layers. We can conclude that the models w/ detach consistently gain the better performance than those with gradient back-propagation (i.e., models w/o detach). Our explanation is that a Max Pooling Layer is employed in the extra branch on middle layers, which could disturb gradients backward from the Average Pooling in the main branch. In the remaining experiments, we all use the models without gradients backward from mid-layers to the main backbone.

3) Evaluation on Inter-Class Relation Augmentation: One more experiment is conducted to verify the effectiveness of our proposed inter-class relation augmentation on the CUB-200-2011, the Stanford Cars, and the FGVC-Aircraft datasets with the backbone of ResNet50+mid, ResNet101+mid and DenseNet161+mid. We explore two types of the Mixup and SnapMix data augmentation method as explained in Sec. III-C. Both methods share the identical structure with the only difference lying on hyper parameter $\alpha$ for weighting the proportion of two images. As shown in Table XI, the first row of each block is our method based on different backbone without using inter-class relation augmentation, while the remaining three are enhanced with inter-class relation augmentation based on the Mixup, the CutMix and the SnapMix, respectively. It is observed that our proposed inter-class relation augmentation can further improve classification performance on all datasets with different backbones. Moreover, our DTRG
with the SnapMix based augmentation can perform slightly better than the ones using the Mixup and the CutMix, owing to its usage of the more precise semantic-relatedness proportion (i.e., $\lambda$ in Eq. (14)) of mixed images.

E. Visualization

To illustrate the rationale of our proposed non-parametric online center loss and graph similarity constraint modules, we utilize the t-SNE [72] to project the features from the last block of trained models (e.g., models based on the ResNet-50 backbone in Fig. 2 (a)) into 2d space. As shown in Fig. 7, the features using our OCL loss (for minimizing intra-class variations) can be more compact and thus distinguishable than the baseline, whereas the features from model with the GSL loss only (for exploiting inter-class relation) can be more structurally distributed along a principle direction than the baseline. Feature centers of each category are generated by the mean of features of samples from the same category, which can be viewed as a simple yet robust prototype of all samples within each class. Intra-class similarity in feature space can be achieved by minimizing average distance to feature centers from all data points of each category. What’s more, the proposed target relation graph between different feature centers can serve as additional supervision signals to incorporate inter-class relation into representation learning, which thus can produce a structurally distributed feature space with limited principle directions (affected by highly-correlated neighbouring classes). As shown in Fig. 7(b), visual representations of each class will be more compact via the proposed online center loss, while with the constraint of our Target Relation Graph, intra-class samples can be more structurally distributed along a principle direction, as shown in Fig. 7(c). Using our method, the feature centers of each class are updated in an iterative manner, which will affect generation of target relation graph. Therefore, inter-class correlation can be dynamically varying during model training. However, our DTRG method on enforcing the model to discover visual patterns across classes can encourage better representations.

To show visual patterns captured by the proposed DTRG, we compare the attention maps of our DTRG with the SnapMix [22] method on the CUB-200-2011 dataset. Visualization based on the Gradient CAMs [73] are displayed in Fig. 8. It is observed that the SnapMix method depends on features extracted from limited discriminative regions, while our DTRG can enforce the model to focus on not only discriminative regions, but also less informative parts of objects guided by inter-class semantic correlation. Visualized results demonstrate that our method can improve representations by capturing discriminative information from less informative regions. Consequently, it is proven that our DTRG method can encourage more discriminative features.

V. CONCLUSION

In this paper, we propose a self-supervised feature regularization method on deep representation learning for visual classification, which can incorporate inter-class correlation into features and also mitigate intra-class feature variations. Superior performance can further be achieved via the proposed inter-class relation augmentation. Extensive experiments on popular benchmarks of multiple visual classification tasks can demonstrate the effectiveness of each proposed modules, consistently achieving the state-of-the-art performance. Moreover, our method can be more robust against sparse and long-tail data distributions owing to exploiting latent target relation.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 25, 2012, pp. 1–9.
[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4700–4708.
[4] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltech-ucsd birds-200-2011 dataset,” Inst. Technol., Pasadena, CA, USA, Tech. Rep. 2010-001, 2011.
[5] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, “Novel dataset for fine-grained image categorization: Stanford dogs,” in Proc. CVPR Workshop Fine-Grained Vis. Categorization, 2011, vol. 2, no. 1, pp. 1–2.
[6] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3D object representations for fine-grained categorization,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, Dec. 2013, pp. 554–561.
[7] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, “Fine-grained visual classification of aircraft,” 2013, arXiv:1306.5151.
[8] S. Reed, Z. Akata, H. Lee, and B. Schiele, “Learning deep representations of fine-grained visual descriptions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 49–58.
[9] T. Chen, W. Wu, Y. Gao, L. Dong, X. Luo, and L. Lin, “Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding,” in Proc. 26th ACM Int. Conf. Multimedia, Oct. 2018, pp. 2023–2031.
[10] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, “Hierarchical part matching for fine-grained visual categorization,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 1641–1648.
[11] N. Zhong, J. Donahue, R. Girshick, and T. Darrell, “Part-based R-CNNs for fine-grained category detection,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 834–849.
[12] S. Branson, G. Van Horn, S. Belongie, and P. Perona, “Bird species categorization using pose normalized deep convolutional nets,” 2014, arXiv:1406.2952.
[61] Y. Song, N. Sebe, and W. Wang, “Why approximate matrix square root outperforms accurate SVD in global covariance pooling?” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 1115–1123.
[62] T. Yu, X. Li, and P. Li, “Fast and compact bilinear pooling by shifted random Maclaurin,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 3243–3251.
[63] Q. Wang, J. Xie, W. Zou, L. Zhang, and P. Li, “Deep CNNs meet global covariance pooling: Better representation and generalization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 8, pp. 2582–2597, Aug. 2021.
[64] T. Yu, Y. Cai, and P. Li, “Efficient compact bilinear pooling via Kronecker product,” in Proc. AAAI Conf. Artif. Intell., Jun. 2022, pp. 3170–3178.
[65] P. Koniusz and H. Zhang, “Power normalizations in fine-grained image, R. Du T. Yu, X. Li, and P. Li, “Fast and compact bilinear pooling by
[66] 5584 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 31, 2022
[67] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and
[68] S. Lee and B. C. Song, “Graph-based knowledge distillation by multi-
[69] Y. Liu et al.
[70] H. Chen, Y. Wang, C. Xu, C. Xu, and D. Tao, “Learning Student
[71] networks via feature embedding,” IEEE Trans. Neural Netw. Learn.
[72] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.
[73] G. Van Horn et al., “The iNaturalist species classification and detection dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8769–8778.
[74] A. Krizhevsky, “Learning multiple layers of features from tiny images,” M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, CA, USA, 2009.
[75] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.
[76] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, “Describing textures in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 3606–3613.
[77] A. Quattioni and A. Torralba, “Recognizing indoor scenes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 413–420.
[78] S. Park, J. Lim, Y. Jeon, and J. Y. Choi, “Influence-balanced loss for imbalanced visual classification,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2021, pp. 735–744.
[79] Y. Wang, V. I. Morariu, and L. S. Davis, “Learning a discriminative filter bank within a CNN for fine-grained recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4148–4157.
[80] Y. Zhao, K. Yan, F. Huang, and J. Li, “Graph-based high-order relation discovery for fine-grained recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 15079–15088.
[81] S. Huang, X. Wang, and D. Tao, “Stochastic partial swap: Enhanced model generalization and interpretability for fine-grained recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 620–629.
[82] Y. Gao, X. Han, X. Wang, W. Huang, and M. Scott, “Channel interaction networks for fine-grained image categorization,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, no. 7, pp. 10818–10825.
[83] C. Liu, H. Xie, Z.-J. Zha, L. Ma, L. Yu, and Y. Zhang, “Filtration and distillation: Enhancing region attention for fine-grained visual categorization,” in Proc. AAAI Conf. Artif. Intell., Apr. 2020, pp. 11555–11562.
[84] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9268–9277.
[85] K. Cao, C. Wei, A. Gaidon, N. Areechiga, and T. Ma, “Learning imbalanced datasets with label-distribution-aware margin loss,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–12.
[86] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 2980–2988.
[87] S. Rahman, L. Wang, C. Sun, and L. Zhou, “ReDro: Efficiently learning large-sized SPD visual representation,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 1–17.
[88] M. Buda, A. Mak, and M. A. Mazzurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” Neural Netw., vol. 106, pp. 249–259, Oct. 2017.

Ke Chen (Member, IEEE) received the B.E. degree in automation and the M.E. degree in software engineering from Sun Yat-sen University in 2007 and 2009, respectively, and the Ph.D. degree in computer vision from the Queen Mary University of London in 2013. He is currently an Associate Professor with the School of Electronic and Information Engineering, South China University of Technology. His research interests are in computer vision, deep learning, and pattern recognition.

Kui Jia (Member, IEEE) received the B.E. degree from Northwestern Polytechnical University, Xi’an, China, in 2001, the M.E. degree from the National University of Singapore, Singapore, in 2004, and the Ph.D. degree in computer science from the Queen Mary University of London, London, U.K., in 2007. He was with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, The Chinese University of Hong Kong, Hong Kong, the Institute of Advanced Studies, University of Illinois at Urbana-Champaign, Champaign, IL, USA, and the University of Macau, Macau, China. He is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. His recent research focuses on theoretical deep learning and its applications in vision and robotic problems, including deep learning of 3D data and deep transfer learning.

Kanjin Liu received the B.E. degree from the College of Mechanical and Vehicle Engineering, Hunan University, China, in 2016. He is currently pursuing the Ph.D. degree with the Shien-Ming Wu School of Intelligent Engineering, South China University of Technology. His research interests are in computer vision, deep learning, and pattern recognition.

Hong Kong, Hong Kong, the Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, in 2007. He was with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, in 2001, the M.E. degree in software engineering from Sun Yat-sen University in 2007 and 2009, respectively, and the Ph.D. degree in computer vision from the Queen Mary University of London in 2013. He is currently an Associate Professor with the School of Electronic and Information Engineering, South China University of Technology (SCUT). Before joining SCUT, he was a Postdoctoral Research Fellow with the Department of Signal Processing, Tampere University of Technology. His research interests include computer vision, pattern recognition, neural dynamic modeling, and robotic inverse kinematics.