Rethinking Batch Sample Relationships for Data Representation: A Batch-Graph Transformer Based Approach

Xixi Wang, Bo Jiang, Xiao Wang, Member, IEEE, Jinhui Tang, Senior Member, IEEE, and Bin Luo, Senior Member, IEEE

Abstract—Exploring sample relationships within each mini-batch has shown great potential for learning image representations. Existing works generally adopt the regular Transformer to model the visual content relationships, ignoring the cues of semantic/label correlations between samples. Also, they generally adopt the full self-attention mechanism which are obviously redundant and also sensitive to the noisy samples. To overcome these issues, in this paper, we design a simple yet flexible Batch-Graph Transformer (BGFormer) for mini-batch sample representations by deeply capturing the relationships of image samples from both visual and semantic perspectives. BGFormer has three main aspects. (1) It employs a flexible graph model, termed Batch Graph to jointly encode both visual and semantic relationships of samples within each mini-batch. (2) It explores the neighborhood relationships of samples by borrowing the idea of sparse graph representation which thus performs robustly, w.r.t., noisy samples. (3) It devises a novel specific Transformer architecture that mainly adopts dual structure-constrained self-attention (SSA), together with graph normalization, FFN, etc, to carefully exploit the batch graph information for sample tokens (nodes) representations. As an application, we apply BGFormer to the metric learning tasks. Extensive experiments on four popular datasets demonstrate the effectiveness of the proposed model.

Index Terms—Transformer, mini-batch, graph model, metric learning.

I. INTRODUCTION

LEARNING robust and discriminative image feature representation is a fundamental task in computer vision applications, such as image classification [1], [2], visual recognition [3], [4], object tracking [5], [6], metric learning [7], [8], [9] and so on. With the development of deep learning, Convolutional Neural Networks (CNNs) have been widely used for image representation learning. However, CNNs just model the local information of images and thus fail to fully capture the global context information of images. It is well known that Transformers [10], [11] have achieved great successes in many visual representation and learning tasks due to their core self-attention mechanism to capture the long-range dependencies among visual units, e.g., patches, regions, etc. In recent years, various Transformer variants [12], [13], [14], [15] have been studied in computer vision task to facilitate the better image feature representation learning problem. For example, Dosovitskiy et al. [2] develop a Visual Transformer (ViT) architecture to model the contextual relationships of image regions and thus obtain the competitive image classification results. Liu et al. [11] propose a hierarchical vision Transformer with shifted windows, namely Swin Transformer, to produce the hierarchical feature representation by using hierarchical architecture. Rao et al. [16] propose a DynamicViT model to obtain the better classification results by dynamically pruning redundant image patches/regions. Chen et al. [5] introduce a Transformer Tracking (TransT) to combine template and search region features by using attention for object tracking task. Han et al. [12] develop a Vision GNN (ViG) architecture to generate graph-level feature by viewing the patches of image as nodes for visual tasks. He et al. [17] introduce dense classifier and attentive pooling for few-shot classification, which employs attention regressor to focus on the more important local descriptors and then utilizes the pooling operation for producing image-level embeddings.

Beyond exploring local patch/region relationships for image representation, recent studies also propose to leverage Transformer to explore sample-level correlations for image representation learning [15], [18], [19], [20]. One representative work is BatchFormer [15], [20] which aims to capture the interaction between all samples in each mini-batch to learn robust data representations. Mondal et al. [21] propose to employ GNN for batch sample learning. Wang et al. [18] recently propose QSFFormer to deeply explore the interactions between query and support samples for few-shot classification problem. However, existing models generally directly adopt the ideas of traditional Transformers [2], [10]. Also, they generally only focus on modeling the visual relationships among images, ignoring the important cues of semantic/label correlations among images. Moreover,
existing BatchFormer generally adopts regular self-attention (SA) mechanism which learns each image by aggregating the information from all other samples. Obviously, this mechanism is redundant and also sensitive to the noisy samples.

To overcome these limitations, in this article, we rethink to explore sample relationships and propose a novel Batch-Graph Transformer (BGFormer). The proposed BGFormer is simple and provides a general plug-and-play module for data representation and learning tasks. The key aspects of the BGFormer are twofold. First, it employs a flexible graph model, termed Batch Graph to deeply model the relationships of samples within each mini-batch. In particular, the proposed batch graph encodes both visual and semantic/label relationships of samples simultaneously in a unified model, as illustrated in Fig. 1. Also, instead of utilizing the information of full relationships, our batch graph explores neighborhood relationships among samples by borrowing the idea of sparse graph representation [22], [23], [24] which thus perform robustly, w.r.t., noisy samples. Second, based on the built batch graph, we then develop a novel graph learning mechanism, termed Batch-Graph Transformer (BGFormer), which aims to learn contextual representations for samples in a mini-batch by conducting message passing on the batch graph, as illustrated in Fig. 2. The core of BGFormer block is the dual Structure-constrained Self-attention (SSA) to simultaneously capture visual and semantic relationships for data representation.

To validate the effectiveness of the proposed BGFormer, we apply it to the metric learning tasks and develop a new deep metric learning approach. As shown in Fig. 3, our proposed metric learning framework is composed of backbone network, BGFormer module, hyperbolic embedding module and metric learning. We first utilize the backbone network to extract the initial features for data samples. Then, the proposed BGFormer is introduced to learn the more discriminative and richer representations by capturing the dependence of data samples. Finally, we utilize the hyperbolic embedding module with shared parameters to map the learned sample features into hyperbolic space for metric learning tasks, as used in work [8], [15].

Overall, the main contributions of this work are summarized as follows:

- We rethink to deeply explore sample relationships and propose a novel batch graph to encode both visual and label interactions of different samples in each mini-batch.
- We develop a novel specific BGFormer which conducts message propagation across batch graph to capture the relationships of samples. Different from traditional Transformers, the key aspect of BGFormer is our new derivation of dual Structure-constrained Self-Attention (SSA) mechanism to achieve the message propagation on the batch graph.
- We integrate BGFormer into a deep metric learning scheme and propose an effective end-to-end metric learning approach. Extensive experiments on four public popular datasets demonstrate the effectiveness and superiority of the proposed BGFormer.

II. RELATED WORKS

A. Vision Transformer

Since the Transformer architecture is proposed by Vaswani et al. [10] in 2017, it has been widely used in the field of computer vision (CV) due to its excellent modeling ability, including image classification [11], [16], [18], object re-identification [3], [4], [25], [26], object detection [27], [28], object tracking [5], [6], etc. For example, Carion et al. [27] propose a Transformer encoder-decoder architecture, termed DETransformer (DETR for short), which drops multiple hand-designed components and builds the fully end-to-end object detector for object detection task. Liu et al. [11] design
a hierarchical vision Transformer, which generates the feature representation with the shifted windows for image classification and other vision tasks. He et al. [4] introduce a pure Transformer-based object ReID framework, named TransReID, which splits each image as a sequence of patches and then feeds into TransReID to encode the feature representations for object ReID task. Wang et al. [29] design a Siamese-like tracking pipeline based on the transformer architecture, which use two parallel branches, i.e., encoder branch and decoder branch, to bridge search and template frame for object tracking task. Zhang et al. [30] propose a Transformer-based backbone network based on dynamic graph message passing network for classification pretraining, object detection, instance and semantic segmentation. Tang et al. [31] develop a transformer-based model GEN-OBT which contains gloss encoder, pose decoder and online reverse gloss decoder for sign language production task. The above works generally adopt the Transformer as the enhanced feature extractor to obtain better performance.

On the other hand, some metric learning [8], [32] models are also developed based on the Transformer network. Specifically, El-Nouby et al. [32] directly introduce the Transformer architecture to obtain the image feature representation for metric learning task. Ermolov et al. [8] propose to combine the Transformer network and hyperbolic distance for obtaining the better results of metric learning. These works divide an image into a sequence of patches as the input of Transformer, but only model the contextual correlation of intra-image well. Seldom existing algorithms consider modeling relationships of inter-images using Transformer architecture. In this work, we propose a novel image-level Transformer network, termed Batch-Graph Transformer (BGFormer) which captures the visual and semantic/label relationship between all the input images in each mini-batch by constructing batch graph. Therefore, we can achieve better feature representation learning for high-performance classification, such as over-reliance on hand-crafted distance metrics, and lack of robustness when dealing with imbalanced or noisy data. To solve these problems, in recent years, there are many methods which can be briefly summarized into two categories: distance metric learning and deep metric learning. The distance metric learning [39], [40], [41], [42], [43], [44], [45] aims to design various loss functions to constrain the image sample of same class to be closer and different classes to be farther, including contrastive loss, normalized softmax loss, adversarial loss, hierarchical triplet loss, softTriple loss, proxy-anchor loss and so on. Unlike it, the goal of deep metric learning [33], [34], [35], [36], [37], [46] is usually to obtain better data sample feature embeddings through various techniques or model architectures. For example, Wu et al. [33] propose a distance weighted sampling strategy to uniformly extract samples according to the relative distance between samples for better embedding representation. Kim et al. [34] develop an attention-based ensemble based on multiple attention mask to improve the diverse of learners in feature embeddings for deep metric learning. Roth et al. [35] propose a model containing two encoder to explicitly learn the latent features shared between object classes in order to improve the performance of metric learning. Wang et al. [36] propose a cross-batch memory (XBM) mechanism to update the image features of recent mini-batches by memorizes the image embeddings of past mini-batches. Although good results can be obtained, however, these works adopt the CNN as their feature extractor which may obtain sub-optimal results only. In this work, we demonstrate the effectiveness of Transformer architecture by conducting experimental analysis and extend it to Batch-Graph Transformer to explore the sample relationship among all the input images in each mini-batch for more robust and discriminative feature representations.

C. Graph Representation Learning

The goal of graph representation learning [22], [23], [47], [48], [49] is to project the nodes of graph to vector representations and retain as much topological information as possible. In early years, local sparse and low-rank graph shallow learning models have been widely studied [22], [23], [24], such as, Jiang et al. [24] propose Graph Laplacian PCA by incorporate...
local $k$-NN graph learning into PCA. In recent years, graph neural networks (GNNs) and graph Transformers have been studied. For example, Kipf et al. [50] propose Graph Convolutional Network (GCN), which applies convolution operation to graphs and models the local structure and node features of learning graphs through approximate analysis in the graph spectral domain. Velickovic et al. [47] develop graph attention network (GAT), which computes the hidden representations of each node in the graph and follows a self-attention strategy to aggregate its neighbors. Jiang et al. [51] propose a Graph elastic Convolutional Network (GeCN) for local sparse graph data representation learning. Ying et al. [48] develop a Graphformer with centrality and spatial encodings to obtain the good results on graph representation learning. Chen et al. [49] propose a Structure-Aware Transformer to incorporate the structural information into the original self-attention.

In recent years, good performance has also been obtained by employing GCNs/GNNs on various computer vision tasks. For example, Qi et al. [52] propose a Graph Parsing Neural Network (GPNN) based on the message passing neural network to provide a HIO representation for spatial and spatial-temporal domains. Gao et al. [53] design a Graph Convolutional Tracking (GCT) to model structured representation of historical target exemplars for visual tracking. Chen et al. [54] propose a multi-label recognition model based on Graph Convolutional Networks to learn inter-dependent class-level representation. Mondal et al. [21] introduce a MBGNN with Graph Neural Networks by constructs a kNN graph only representing the visual relationships of samples for a downstream classification task. Xie et al. [55] develop a scale-aware graph neural network (SAGNN) to learn the cross-scale relations between support-query image pairs for few-shot semantic segmentation. Tang et al. [56] propose a graph-based multimodal sequential embedding network by introducing a graph embedding unit that embeds parallel channel-wise and temporal-wise convolutions into graph convolution to achieve the inter-modal and intra-modal correlations for sign language translation task. Differently, in this work, we introduce batch graph to model both visual and semantic/label relationships between all the input images in each mini-batch and develop a Batch-Graph Transformer for image-level representation and learning.

### III. METHODOLOGY

In this section, we propose a novel Batch-graph Transformer (BGFormer) model to deeply capture the relationships of samples in the mini-batch for data representation. BGFormer has two aspects, i.e., i) batch graph construction and ii) representation learning on batch graph, as introduced below in detail.

#### A. Building Batch Graph

To fully encode the relationships of samples in each mini-batch, we build a dual-relational graph $G(V, E^v, E^l)$ which encodes both visual and label correlations among different data samples, as illustrated in Fig. 1. Comparing with previous BatchFormer [15] and MBGNN [21], the proposed batch graph deeply explores sample relationships within mini-batch in both visual and semantic spaces. More details are as follows.

**Nodes:** Each node $v_i$ in the batch graph $G(V, E^v, E^l)$ denotes each sample in the mini-batch. Let $f_i \in \mathbb{R}_C^l$ denotes the initial CNN feature descriptor for sample node $v_i \in V$. Let $F = \{f_1, f_2, \ldots, f_B\} \in \mathbb{R}_B\times C$ represents the collection of node’s initial features, where $B = |V|$ is the total number of samples (i.e., batch size) in a mini-batch size and $C$ denotes the channel number of each sample image. In our experiments, we empirically set the batch size to 900 and thus we have $|V| = 900$.

**Edges:** The edge set in our batch graph involves two types of edges, i.e., $E^v$ encodes the visual content relationships and $E^l$ represents the label relationships among samples. For visual-type edges $E^v$, we adopt the neighboring connection strategy. To be specific, we first learn the affinities/similarities among data samples as $S = FF^T$ as suggested in the simplified self-attention [57]. Then, each node $v_i \in V$ connects to the node $v_j \in V$ that are belong to its $k$ nearest neighborhood, i.e., $v_j \in N_k(v_i)$, as shown in Fig. 1. We can use the weight adjacency matrix $A^v \in \mathbb{R}_B\times B$ to represent visual-type edge connections which is defined as follows,

$$A^v_{ij} = \begin{cases} s_{ij}, & \text{if } v_j \in N_k(v_i) \\ 0, & \text{otherwise}. \end{cases} \quad (1)$$

where $N_k$ denotes the $k$ nearest neighborhoods of node $v_i$. The $s_{ij}$ means the affinity/similarity value between $i$-th sample and $j$-th sample in a mini-batch.

In addition to visual correlations, we also represent the label correlations of data samples via label-type edge set $E^l$. To be specific, each edge $e_{ij} \in E^l$ connects node $v_i$ and $v_j$ if they have the identical class label. We can use an adjacency matrix $A^l \in \mathbb{R}_B\times B$ to encode the label edge connections which is defined as

$$A^l_{ij} = \begin{cases} 1, & \text{if } l(v_i) = l(v_j) \\ 0, & \text{otherwise}. \end{cases} \quad (2)$$

where $l(v_i)$ denotes the label of sample node $v_i$.

#### B. Batch-Graph Transformer

Overall, the proposed BGFormer module involves multiple BGFormer encoders and a layer-norm (LN) is used between different blocks, as shown in Fig. 2. Different from regular Transformer encoder, our BGFormer encoder first leverages two structure-constrained self-attention blocks to propagate information across neighboring samples (tokens) and then fuses them together, followed by LN, FFN and residual operation to output the final data representations.

Specifically, based on the above batch graph representation $\{F, A^v, A^l\}$, we first conduct Structure-constrained Self-attention (SSA) on batch graph as

$$\tilde{F}^v = F + \text{Norm}(A^v)FW \quad (3)$$

$$\tilde{F}^l = F + \text{Norm}(A^l)FW \quad (4)$$

Some more compact sparse graphs [23], [51] can also be adopted here.
where $\text{Norm}(\cdot)$ denotes the graph normalization operation, such as linear normalization, softmax and symmetric Laplacian normalization [50]. $W$ denotes the weight of linear projection which is shared on two SSA blocks to achieve information communication across two blocks. In this article, for visual affinity $A^v$, we use the linear normalization as follows,

$$\text{Norm}(A^v) = \frac{A^v_i - \min_{j \neq i} \{A^v_{ij}\}}{\max_{j \neq i} \{A^v_{ij}\} - \min_{j \neq i} \{A^v_{ij}\}}$$  \hspace{1cm} (5)$$

For label affinity $A^l$, we use the symmetric Laplacian normalization as

$$\text{Norm}(A^l) = D^{-\frac{1}{2}} (A^l + I) D^{-\frac{1}{2}}$$  \hspace{1cm} (6)$$

where $I$ denotes the identity matrix and $D$ is a diagonal matrix with $D_{ii} = \sum_j A^l_{ij}$. Note that the visual graph $A^v$ is obtained by calculating correlation between visual features. To eliminate scaling issue, we employ linear normalization for it in (5). The semantic graph $A^l$ is a binary graph with element being 0 or 1, which only contains the structure information of graph. To capture this information, inspired by [50], we use the symmetric Laplacian normalization for $A^l$ in (6).

After the above SSA module, we then fuse them together by simply combing them as

$$\hat{F} = \lambda \hat{F}^v + (1 - \lambda) \hat{F}^l$$  \hspace{1cm} (7)$$

where $\lambda \in \{0, 1\}$ denotes the balanced hyper-parameter. Finally, we introduce residual connection and feed-forward network (FFN) on $\hat{F}$ to output the final data representation as follows,

$$\hat{F} = \hat{F} + \text{FFN}(\text{LN}(\hat{F}))$$  \hspace{1cm} (8)$$

where $\text{LN}(\cdot)$ denote layer normalization and $\text{FFN}(\cdot)$ is composed of two fully-connected layers.

IV. APPLICATION: METRIC LEARNING

To verify the effectiveness of our proposed BGFormer, we apply it on the metric learning tasks. As shown in Fig. 3, the BGFormer-based metric learning framework mainly consists of backbone network, BGFormer module, hyperbolic embedding module and metric learning.

Let $X = \{(x_i, y_i)\}_{i=1}^B$ be the inputs of backbone network, where $y_i$ is the corresponding label of image $x_i$ and $B$ denotes the number (i.e., batch size) of images in each mini-batch. The backbone network is a feature extraction network, which can be CNNs [1], [58] (e.g., ResNet-50) or Transformer [2], [13], [14] (e.g., ViT, DeiT and DINO). In this work, we employ ViT-S [2] with adding the global average pooling (GAP) layer to the backbone network to generate the initial image feature descriptors $F \in \mathbb{R}^{B \times C}$. Its parameters are initialized by using the pre-trained model on ImageNet-21K [59] for classification tasks. Then, we feed the initial image features $F$ into the proposed BGFormer module to fully explore the relationships of images in each mini-batch for learning the enhanced representations $\hat{F} \in \mathbb{R}^{B \times C}$. After that, based on the image features $F$ and $\hat{F}$, we use the Hyperbolic Embedding module (HEM) involving fully-connected (FC) layer and hyperbolic mapping, which embeds them into the hyperbolic space to calculate the hyperbolic distances between pairs of samples [8]. Finally, we leverage the pairwise cross-entropy loss function which defined on pair-wise distance [8] to achieve the metric learning task.

Since the BGFormer module is not utilized in the testing stage, to transfer the knowledge of training process to the test stage, we derive two Hyperbolic Embedding (HEM) and metric learning branches for the outputs $F$ and $\hat{F}$ respectively, as suggested in work [15]. Both learning branches share the same parameters to achieve the knowledge communication. Let $L_1$ and $L_2$ denote the pairwise cross-entropy loss for these two branches. Then, the total loss of our method is formulated as

$$L_{total} = \alpha L_1 + (1 - \alpha) L_2$$  \hspace{1cm} (9)$$

where $\alpha \in \{0, 1\}$ is a balanced hyper-parameter.

V. EXPERIMENTS

A. Datasets and Evaluation Metrics

Datasets: We evaluate the proposed BGFormer based metric learning method on four public datasets, including CUB-200-2011 (CUB) [60] (200 classes, 11788 images), Cars-196 (Cars) [61] (196 classes, 16185 images), Stanford Online Product (SOP) [62] (22634 classes, 120053 images) and In-shop Clothes Retrieval (In-Shop) [63] (7986 classes, 52712 images), which are all splitted into training set and testing set. Concretely, following works [8], [32], we select 100 classes (3864 images), 98 classes (8054 images) and 11318 classes (39551 images) from CUB, Cars and SOP respectively for training and the rest classes are used for testing. For the In-Shop dataset, we select the first 3997 classes (25882 images) for training and split the remaining 3985 classes into a query set (14218 images) and a gallery set (12612 images) for testing.

Evaluation Metrics: In this work, we utilize the widely adopted Recall@K evaluation metric to compare the proposed method with previous metric learning methods. The Recall@K represents the ratio of the number of positive samples retrieved in the top-K results to the number of all positive samples in the testing set. For a fair comparison, following these works [8], [32], [36], [40], [64], for CUB and Cars, we set $K$ to 1, 2, 4 and 8. For SOP, we set $K$ to 1, 10, 100 and 1000. For In-Shop, we set $K$ to 1, 10, 20 and 30.

B. Implementation Details

In this work, we employ ViT-S [2] network pre-trained on the large-scale ImageNet-21k [59] as the backbone network for extracting the initial image feature representations. We execute our proposed model with Python and train the whole network in an end-to-end way.

The resolution of all images in the CUB dataset is set as 256 $\times$ 256. Following the work [8], other datasets are set as 224 $\times$ 224. In the training phase, we randomly select 100 classes and 9 samples per class in each mini-batch, i.e., batch size $B = 900$. Different learning rates are set for the backbone network and proposed BGFormer module, i.e., 3e-5 and 7.5e-4 respectively. The AdamW [67] is adopted as optimizer, with the
TABLE I
RECALL@K METRIC FOR FOUR DATASETS, WHERE "DIM" COLUMN REPRESENTS THE DIMENSIONALITY OF THE FINAL EMBEDDINGS OF MODEL. THE DIFFERENT BACKBONE NETWORK IS USED BY THESE METHODS, SUCH AS ResNet-50 [1] (NSoftmax, Margin, FastAP, MFC), GoogleNet [65] (SM, A-BIER, ABE), Inception [56] (HTL, SoftTriple, MS, XBM, Proxy-Anchor), and Transformer [2] [32] (HyP-DINO, HyP-ViT).

| Method       | dim | CUB-200-2011 (K) | Cars-196 (K) | SOP (K) | In-Shop (K) |
|--------------|-----|-----------------|--------------|---------|-------------|
| SM [42]      | 512 | 56.0            | 68.3         | 78.2    | 86.3        |
| HTL [42]     | 512 | 57.1            | 68.8         | 78.7    | 86.5        |
| A-BIER [41]  | 512 | 57.5            | 68.7         | 78.3    | 86.2        |
| ABE [34]     | 512 | 60.6            | 73.1         | 82.0    | 90.0        |
| ResNet50 [40]  | 512 | 61.3            | 73.9         | 83.5    | 90.0        |
| SoftTriple [43] | 512 | 55.4            | 74.6         | 84.5    | 90.9        |
| MS [44]      | 512 | 65.7            | 77.0         | 86.3    | 91.2        |
| XBM [36]     | 512 | 65.8            | 77.9         | 84.0    | 89.9        |
| Proxy-Anchor [45] | 512 | 68.4            | 79.2         | 86.8    | 91.6        |

The bold and underline denote the best two results.

TABLE II
COMPARISON RESULTS OF DIFFERENT BACKBONE NETWORKS ON FOUR DATASETS OF METRIC LEARNING TASK.

| Methods | CUB-200-2011 (K) | Cars-196 (K) | SOP (K) | In-Shop (K) |
|---------|-----------------|--------------|---------|-------------|
| ResNet50 | 42.5            | 56.4         | 69.3    | 80.4        |
| DeiT-S   | 74.9            | 83.9         | 93.6    | 96.0        |
| DINO     | 79.5            | 87.4         | 92.5    | 95.4        |
| ViT*     | 84.7            | 90.9         | 94.2    | 96.3        |
| ResNet50 | 45.2            | 58.8         | 71.6    | 81.6        |
| DeiT-S   | 78.2            | 86.6         | 91.8    | 95.2        |
| DINO     | 80.1            | 87.9         | 92.8    | 95.4        |
| ViT*     | 86.6            | 92.0         | 94.9    | 95.7        |

C. Comparison With State-of-The-Art Methods

The comparison results on CUB-200-2011 (CUB) [60], Cars-196 (Cars) [61], Stanford Online Product (SOP) [62] and In-shop Clothes Retrieval (In-Shop) [63] datasets are summarized in Table I. We can clearly see that our proposed model outperforms all current state-of-the-art methods including CNN-based methods [7], [33], [34], [35], [36], [40], [41], [42], [43], [44], [45], [64] and Transformer methods [8], [32]. Note that our proposed approach achieves the best scores on all four settings on the CUB dataset. Specifically, when we set the dimensionality of the final output embeddings as 128, our model improves the weight decay 5e-5 to optimize the whole network. Our model can be trained well with a total number of 400 epochs, and we test the checkpoint every 5 epochs. The checkpoint with the best results is saved as the final model.

From the experimental results on the Cars dataset, we can find that our model achieves the second-best results except for Recall@1. To be specific, compared with [8], our proposed method beats it by +3.3% in Recall@1 and +2.2% in Recall@2. The IRT [32] is also developed based on Transformer architecture, however, we exceed this approach by a large margin, i.e., +10.0% in Recall@1 and +7.0% in Recall@2. Compared with the Proxy-Anchor [45] with the SOTA performance based on CNN architecture, we still obtain +18.2% and +12.8% improvements in Recall@1 and Recall@2 respectively.

From the experimental results on the Cars dataset, we can find that our model achieves the second-best results except for Recall@1. To be specific, compared with [8], our proposed method beats it by +3.3% in Recall@1 and +2.2% in Recall@2 under the same backbone network (i.e., HyP-ViT). HyP-DINO [8] with self-supervised learning and knowledge distillation in the backbone network has better model performance. Similar findings and conclusions can be found in the SOP and InShop datasets. In addition, it is worth noting that our proposed

Hyp-ViT [8] based on Transformer architecture and hyperbolic distance by +2.6% in Recall@1 and +1.8% in Recall@2. The IRT [32] is also developed based on Transformer architecture, however, we exceed this approach by a large margin, i.e., +10.0% in Recall@1 and +7.0% in Recall@2. Compared with the Proxy-Anchor [45] with the SOTA performance based on CNN architecture, we still obtain +18.2% and +12.8% improvements in Recall@1 and Recall@2 respectively.

TABLE III
ANALYSIS EXPERIMENTS OF DIFFERENT COMPONENTS OF THE PROPOSED MODEL, WHERE "W/O LCG" REPRESENTS THE SEMANTIC/LABEL CORRELATION GRAPH IS REMOVED

| Model | Method | CUB-200-2011(K) | Cars-196(K) | SOP(K) | In-Sho(p) |
|-------|--------|-----------------|-------------|--------|-----------|
| 1     | Baseline | 84.7           | 90.9        | 94.2   | 96.1      |
| 2     | +BatchFormer | 85.2          | 91.1        | 94.5   | 96.6      |
| 3     | +BGFormer | 86.1          | 92.2        | 94.9   | 96.6      |
| 4     | +BGFormer | 86.6          | 92.0        | 94.9   | 96.7      |

The bold denotes the best results.
model with more minor or the same low-dimension embeddings achieves the SOTA performance for all datasets.

Overall, our proposed model obtains the best performance on both more challenging fine-grained datasets like Cars and CUB, and large-scale datasets with enormous classes like SOP and InShop. These experimental results fully demonstrate the effectiveness of the proposed model on learning more discriminative feature representations.

Representation Visualization: As shown in Fig. 4, we show the 2D visualizations of the final image representations learned by the Baseline model [8] and the proposed BGFormer on Cars [61] dataset, respectively. Different colors indicate different classes. Intuitively, we can find that the Baseline model can learn the discriminative image feature representation. It is because of the utilization of powerful ViT-S [2] as the backbone network which is pre-trained on the large-scale ImageNet-21k dataset. However, the ‘outliers’ are common to see from Fig. 4(a) due to the baseline model only models the contextual correlation of intra-image but ignores the importance of the correlation of inter-images. In contrast, our proposed BGFormer significantly alleviates the issues of ‘outliers’ and obtains more discriminative feature representations (the distribution of images is also clearer than our Baseline). This visualization further proves the effectiveness of the proposed BGFormer for learning more discriminative and robust image representations.

Qualitative Visualization: As shown in Fig. 5, we give more visualizations of the top-4 ranking results on the Cars [61] dataset. The image with green boundary represents true positive samples and the red boundary denotes false positive samples. We can find that: 1). more true positives can be found in the proposed BGFormer. 2). the proposed BGFormer can accurately identify the same type of car with different colors, such as the 3-th, 4-th, 5-th, 6-th and 7-th rows, which fully demonstrated the ability of BGFormer to close the intra-class gap. 3). the proposed BGFormer can well distinguish samples between different classes with little change, such as two types of cars in 2-th and 3-th rows, 6-th and 7-th rows. In summary, we can obtain higher accuracy using the proposed BGFormer for the metric learning task.

D. Ablation Study

In this section, we conduct extensive ablation studies to further investigate the influence of different modules or sets, such as embedding size, batch size, neighbor number, different backbone networks, etc.

Analysis of Embedding Size: As we all know, the dimension of feature embedding affects the running speed and the overall accuracy. The larger scale of output dimension may carry more information and brings us better performance, but also leads to inefficiency issue and require more computation cost. In this part, we test different embedding dimensions to investigate their influence on our final performance by setting the dimension $d$ to 384, 256, 128, 64, and 16 respectively. As shown in Table IV, we can find that the performance decreases significantly when $d < 128$ and is slightly improved when $d \geq 128$. Therefore, the embedding dimension $d = 128$ is chosen as our default value in other experiments in this work.

Effect of Batch Size: Different from the standard Transformer [2], [13], [14] which extracts the patch-level features, in this work, we propose the BGFormer (Batch-Graph Transformer) to learn the image-level visual representations. Therefore, the batch size may be a key factor for the final performance. As shown in Table IV, we analyze this parameter on the CUB [60] dataset. It is easy to find that the overall performance benefits from the larger batch size $B$, and will be relatively stable when the batch size $B \geq 800$. When the batch size $B < 400$, the performance will drop slightly. In our experiments, we set $B = 900$ to achieve a good balance between the performance and the hardware limitation. In addition, we add the experiments on BatchFormer [15] under different batch size settings on CUB dataset, and the comparison results on Recall@1 and Recall@2 are shown in Fig. 6. We can observe that the performance of
BatchFormer increases with the increasing of batch size, but it is still lower than that of the proposed BGFormer method.

Effect of Neighbor Number: The neighbor number \( K \) controls the computational complexity of structure-constrained self-attention based on visual graph in the proposed BGFormer module. The sparse operation can reduce the computational complexity of feature aggregation from \( O(B^2C) \) to \( O(BKC) \), where \( K \) is a constant and \( K < B \). To this end, we study the influence of different number of neighbors on four datasets, as shown in Fig. 7. We can see that the model performance increases with the decrease of neighbor number until \( K = 100 \). We speculate that the fewer irrelevant image samples are considered in feature aggregation, which reduces the impact of negative samples and makes the image features learned from the proposed model more discriminative. Therefore, we default set \( K = 100 \) in all our experiments.

Analysis of Backbone Network: As shown in Table II, we report the performance of four public datasets that extract initial image features based on different backbones, including ResNet50 [1], [58], DeiT-S [13], DINO [14], and ViT-S [2]. We can find that 1). the performance of Transformer architecture is obviously better than CNN-based network (i.e., ResNet50). We think the possible reasons for the large performance gap are as follows. It is known that Transformers can capture the long-range dependencies of image units (e.g., patches) which have been demonstrated much more powerfully than local CNN-based models for image representation [2], [27] on various learning tasks, such as classification, detection, etc. Similar to classification task, for metric learning problem, one main issue is how to learn image data representation effectively. That is to say, the effective image representation can be obviously beneficial for metric learning tasks. Therefore, Transformer models generally obviously outperform CNN methods on metric learning tasks, as shown in Table II. 2).
BGFormer with different backbones, i.e., DeiT-S and DINO, and ViT-S, most of our results are also better than the baseline method. 3) our proposed BGFormer achieves the best performance when ViT-S is employed as the backbone network, except for the Cars [61] dataset. However, it achieves sub-optimal performance on the Cars dataset. As a result, ViT-S is adopted as the default backbone of our model in all our experiments.

**Analysis of Different Components:** To verify the effectiveness of each module in our proposed model, we conduct extensive ablation analysis as shown in Table III. To be specific, the baseline (Model 1) adopts the standard ViT-S [2] model for deep metric learning as suggested in [8]. Based on this model, we directly add BatchFormer (Model 2) to model the relationships among samples in each mini-batch by following [15]. Instead of using full relations, we develop BGFormer (w/o LCG) by adding sparse graph representation to avoid the impact of negative samples (Model 3). In addition, we add structure-constrained self-attention based on semantic/label graph, term Model 4.

The following observations can be clearly found from Table III: 1) Compared with Model 1 (baseline), the performance of Model 2 is improved on four datasets, which indicates the effectiveness of modeling the sample correlation. 2) Compared with Model 2, Model 3 significantly improves the performance of baseline (Model 1) on four datasets, showing the effectiveness of our proposed BGFormer (w/o LCG) by reducing the impact of negative samples. 3) Compared with Model 3, Model 4 further improves the performance of BGFormer (w/o LCG), which indicates the effectiveness of dual structure-constrained self-attention (SSA). In summary, the effectiveness and superiority of each component in our proposed model can be fully verified with the experimental results.

**E. Parameter Analysis**

The following three parameters are very important for our model, i.e., number of BGFormer layers $N$, $\alpha$ in (9) and $\lambda$ in (7). Several experiments are conducted on the CUB [60] dataset to check their influence. As shown in Fig. 8, when increasing the number of BGFormer layers $N$, we can find that the model performance increases until $N = 2$. For parameter $\alpha$, as shown in Fig. 9(a), the model performance is continuously improved until $\alpha = 0.6$. For parameter $\lambda$, the performance is continuously improved until $\lambda = 0.4$, as illustrated in Fig. 9(b). As a result, we set $N = 2$, $\alpha = 0.6$, and $\lambda = 0.4$ in our experiments to obtain better results.

VI. CONCLUSION AND FUTURE WORK

In this article, we rethink to explore sample relationships within each mini-batch for discriminative and robust data representation and learning. To this end, we first introduce a novel batch graph to jointly encode both visual and label relationships of samples via a unified model. Then, we develop a novel specific BGFormer to learn sample-level representation. BGFormer adopts the dual SSA mechanism to conduct message passing on the batch graph and fuses both visual and label cues together, followed by FFN and LN operation, to obtain the enrichment data representations. To validate the effectiveness of the proposed BGFormer, we apply it to metric learning tasks and obtain better performance on four benchmark datasets.

One main limitation of our current BGFormer is that the batch graph used in BGFormer is the human established kNN graph which may be sub-optimal to capture the intrinsic relationships of samples within each mini-batch. In our future work, we will employ some graph learning techniques to learn an optimal sparse graph for the proposed BGFormer.
REFERENCES

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

[2] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Representations, 2020.

[3] B. Jiang, X. Wang, A. Zheng, J. Tang, and B. Luo, “Ph-GCN: Person retrieval with part-based hierarchical graph convolutional network,” IEEE Trans. Multimedia, vol. 24, pp. 3218–3228, 2022.

[4] S. He et al., “Transreid: Transformer-based object re-identification,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 15013–15022.

[5] X. Chen et al., “Transformer tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 8126–8135.

[6] Z. Fu, Z. Fu, Q. Liu, W. Cai, and Y. Wang, “SparseTT: Visual tracking with sparse transformers,” in Proc. IEEE Int. Joint Conf. Artif. Intell., 2022, pp. 905–912.

[7] Y. Suh, B. Han, W. Kim, and K. M. Lee, “Stochastic class-based hard example mining for deep metric learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7251–7259.

[8] A. Ermolov, L. Mirvakhabova, V. Khrulkov, N. Sebe, and I. Oseledets, “Hyperbolic vision transformers: Combining improvements in metric learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 7409–7419.

[9] J. Wang, X. Li, Z. Zhang, W. Song, and W. Guo, “Ranked similarity weighting and top-nk sampling in deep metric learning,” IEEE Trans. Multimedia, early access, Nov. 30, 2022, doi: 10.1109/TMM.2022.3225738.

[10] Y. Suh, B. Han, W. Kim, and K. M. Lee, “Stochastic class-based hard example mining for deep metric learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 6000–6010.

[11] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 10012–10022.

[12] K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu, “Vision GNN: An image is worth graph of nodes,” Adv. Neural Inf. Process. Syst., vol. 35, pp. 8291–8303, 2022.

[13] H. Touvron et al., “Training data-efficient image transformers & distillation through attention,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 10347–10357.

[14] M. Caron et al., “Emerging properties in self-supervised vision transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 9650–9660.

[15] Z. Hou, B. Yu, and D. Tao, “Batchformer: Learning to explore sample relationships for robust representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 7256–7266.

[16] Y. Rao et al., “DynamicViT: Efficient vision transformers with dynamic token sparsification,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 13937–13949.

[17] J. He, R. Hong, X. Liu, M. Xu, and Q. Sun, “Revisiting local descriptor for improved few-shot classification,” ACM Trans. Multimedia Comput. Commun., Appl., vol. 18, no. 2s, pp. 1–23, 2022.

[18] X. Wang, X. Wang, B. Jiang, and B. Luo, “Few-shot learning meets transformer: Unified query-support transformers for few-shot classification,” IEEE Trans. Circuits Syst. Video Technol., to be published, doi: 10.1109/TCSVT.2022.6456158.

[19] H. Wang, J. Shen, Y. Liu, Y. Gao, and E. Gavves, “NFormer: Robust person re-identification with neighbor transformer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 7297–7307.

[20] Z. Hou, B. Yu, C. Wang, Y. Zhan, and D. Tao, “BatchFormerV2: Exploring sample relationships for dense representation learning,” 2022, arXiv:2205.07205.

[21] A. K. Mondal, V. Jain, and K. Siddiqi, “Mini-batch graphs for robust image classification,” in Proc. Brit. Mach. Vis. Conf., 2022, pp. 194–206.

[22] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, pp. 2323–2326, 2000.

[23] L. Zhuang et al., “Non-negative low rank and sparse graph for semi-supervised learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2012, pp. 2338–2345.

[24] J. Bo, C. Ding, B. Luo, and J. Tang, “Graph-Laplacian PCA: Closed-form solution and robustness,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2013, pp. 3492C–3498.

[25] J. Zhao et al., “Spatial-channel enhanced transformer for visible-infrared person re-identification,” IEEE Trans. Multimedia, early access, Mar. 31, 2022, doi: 10.1109/TMM.2022.3163847.
Xiao Wang (Member, IEEE) received the B.S. degree from West Anhui University, Luan, China, in 2013 and the Ph.D. degree in computer science from Anhui University, Hefei, China, in 2019. From 2015 to 2016, he was a visiting student with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He also had a visit to UBTech Sydney Artificial Intelligence Centre, Faculty of Engineering, University of Sydney, in 2019. From 2020 to 2022, he finished the postdoc research at Peng Cheng Laboratory, Shenzhen, China. He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. His research interests include computer vision, event-based vision, machine learning, and pattern recognition. He is a Reviewer for a number of journals and conferences, such as IEEE TCSVT, TNNLS, TIP, IJCV, CVIU, PR, CVPR, ICCV, AAAI, ECCV, ACCV, ACM-MM, and WACV.

Jinhui Tang (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2003 and 2008, respectively. He is currently a Professor with the Nanjing University of Science and Technology, Nanjing, China. He has authored more than 200 articles in top-tier journals and conferences. His research interests include multimedia analysis and computer vision. Tang was the recipient of the Best Paper Awards in ACM MM 2007 and ACM MM Asia 2020, Best Paper Runner-Up in ACM MM 2015. He was an Associate Editor for the IEEE TNNLS, IEEE TKDE, IEEE TMM, and IEEE TCSVT. He is a Fellow of IAPR.

Bin Luo (Senior Member, IEEE) received the B.Eng. degree in electronics and M.Eng. degree in computer science from the University of York, York, U.K., in 1991, respectively, and the Ph.D. degree in computer science from the University of York, York, U.K. He has authored or coauthored more than 200 papers in journal and refereed conferences. He is currently a Professor with Anhui University. He Chairs the IEEE Hefei Subsection. He was a peer Reviewer of international academic journals such as, IEEE TRANSACTION ON PAMI, Pattern Recognition, and Pattern Recognition Letters. His research interests include random graph based pattern recognition, image and graph matching, spectral analysis.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.