Distilling Knowledge from Self-Supervised Teacher by Embedding Graph Alignment

Yuchen Ma[*1]
y.yuchenma@gmail.com

Yanbei Chen[*1]
yanbeic@gmail.com

Zeynep Akata[1,2,3]
zeynep.akata@uni-tuebingen.de

[1] University of Tübingen, Germany
[2] MPI for Informatics, Germany
[3] MPI for Intelligent Systems, Germany

Abstract

Recent advances have indicated the strengths of self-supervised pre-training for improving representation learning on downstream tasks. Existing works often utilize self-supervised pre-trained models by fine-tuning on downstream tasks. However, fine-tuning does not generalize to the case when one needs to build a customized model architecture different from the self-supervised model. In this work, we formulate a new knowledge distillation framework to transfer the knowledge from self-supervised pre-trained models to any other student network by a novel approach named Embedding Graph Alignment. Specifically, inspired by the spirit of instance discrimination in self-supervised learning, we model the instance-instance relations by a graph formulation in the feature embedding space and distill the self-supervised teacher knowledge to a student network by aligning the teacher graph and the student graph. Our distillation scheme can be flexibly applied to transfer the self-supervised knowledge to enhance representation learning on various student networks. We demonstrate that our model outperforms multiple representative knowledge distillation methods on three benchmark datasets, including CIFAR100, STL10, and TinyImageNet. Code is here: https://github.com/yccm/EGA.

1 Introduction

Self-supervised learning models are recently shown to be successful unsupervised learners that could greatly boost representation learning on different downstream tasks [5, 7, 8, 15, 21, 32, 33, 35]. By fine-tuning a self-supervised pre-trained model, self-supervised knowledge can often bring more advanced model performance in a wide variety of downstream tasks, such as image classification, object detection, and instance segmentation [9, 15]. However, fine-tuning a self-supervised model could be undesirable in several aspects. First, fine-tuning only generalizes when using the same network architecture for the downstream task, thus requiring self-supervised pre-training to be performed for each customized network. Second, fine-tuning does not allow model compression, as it is infeasible to achieve knowledge transfer to a small lightweight network.
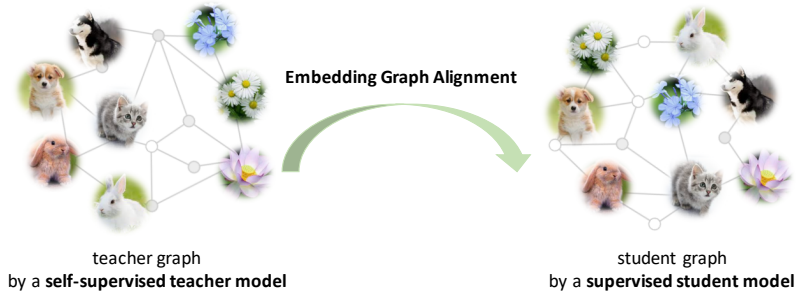
[*]Equal contribution. The work was done when Yuchen Ma was enrolled in the MSc program at the University of Heidelberg.

Embedding Graph Alignment

teacher graph
by a **self-supervised teacher model**

student graph
by a **supervised student model**

Fig. 1: We consider modeling the instance-instance correlations by constructing graphs in the embedding space and transferring the graph structural knowledge from a self-supervised teacher model to a supervised student model by aligning the teacher and student graphs.

In this work, to leverage the strength of self-supervised pre-trained networks, we explore the knowledge distillation paradigm [4, 6, 16] rather than using fine-tuning to transfer knowledge from a self-supervised pre-trained teacher model to a small, lightweight supervised student network. To facilitate research in this regime, we first establish a knowledge distillation benchmark to test knowledge distillation upon various combinations of self-supervised teacher networks and supervised student networks. Given that the most representative self-supervised models are often trained to learn discriminative representations through instance-wise supervision signals [5, 7, 8, 15, 17, 21, 24, 32, 33, 35, 37], we hypothesize that these self-supervised teacher models offer a good structural information among different instances. To distill such fine-grained structural information learned by a self-supervised pre-trained model, we propose to model the instance-instance relationships by a graph structure in the embedding space and distill such the structural information among instances by aligning the teacher graph and the student graph – named as Embedding Graph Alignment (EGA), as shown in Figure 1. Although our approach is being motivated to distill the structure of instance-instance relationships encoded in a self-supervised teacher model, our proposed model can also generalize well to distill knowledge from s supervised teacher model, therefore serving as a generic distillation scheme.

Our contribution is three-fold: (1) We propose a new knowledge distillation method called Embedding Graph Alignment (EGA), which can transfer the instance-wise structural knowledge from a self-supervised teacher model to a supervised student model by aligning the teacher graph and student graph in the embedding space. (2) We establish a comprehensive benchmark on three image classification datasets (CIFAR100, STL10, TinyImageNet) for studying knowledge distillation upon various self-supervised teacher models, comparing our method extensively to multiple state-of-the-art knowledge distillation approaches. (3) We demonstrate the superiority of our model under a variety of evaluation setups, including using different combinations of teacher and student models, using different training strategies in knowledge distillation, and using both self-supervised pre-trained network and supervised pre-trained network as the teacher models.

2 Related Work

Knowledge distillation is a machine learning paradigm that employs a large teacher network or an ensemble of networks to guide the learning of a small, lightweight student net-

work [4, 6, 16]. By mimicking the output distributions or the representations learned by a teacher model, the student network is often regularized with additional supervision signals to achieve superior model generalization, whilst such supervision is often computed as loss functions upon certain training targets. For instance, to mimic the teacher's output distributions, the first knowledge distillation (KD) approach introduces a cross-entropy loss or KL divergence computed on the *soft targets* from a teacher model [16]. To mimic the representations from a teacher model, the latter work FitNet utilizes a regression loss to align the intermediate representations between the teacher and the student networks [30]. The recent following works formulated more advanced and effective loss formulations for knowledge distillation [26, 27, 28, 34, 36], such as aligning representations of attention maps [36], matching the feature distributions with probabilistic models [27], mimicking the similarity scores of feature activations [34], transferring the distance-wise and angle-wise relations [26], or contrasting the representations between teacher and student [32]. Another line of following works explore to distill the knowledge learned from other datasets or data modalities [1, 2, 3, 10, 13, 13], such as distilling the knowledge across the audio and visual data modalities [1, 10]. In contrast to these existing works, we investigate distillation to exploit the knowledge captured from self-supervised models, and formulate a graph structure to encode the fine-grained structural instance-instance relationships learned in a self-supervised model. Our newly proposed Embedding Graph Alignment scheme particularly enables transferring the fine-grained structural semantics from the self-supervised teacher to the student network.

Self-supervised learning (SSL) is a prevalent learning paradigm that drives model training by designing a proxy protocol to construct pseudo label supervision. By formulating unsupervised surrogate losses with free labels, SSL can learn meaningful visual representations in a fully unsupervised manner. One line of works seek for designing self-supervised pretext tasks, including predicting the expected pixel values of an output image [19, 38, 39], solving a surrogate proxy task such as predicting rotations [12], scale and tiling [23], and patch orderings [11, 22, 51]. Another line of works propose to learn discriminative representations through instance-wise supervision signals [5, 7, 8, 15, 17, 21, 24, 32, 33, 35, 37], as most represented by contrastive learning methods [7, 14, 15, 25] such as SimLR [7] and MoCo [15]; whilst the general idea of these works is to enforce instance-wise invariance towards different data augmentation applied on the same input image. More recently, such an instance-wise alignment scheme is further extended for training visual and language models jointly to capture the high-level semantics and learn visual representations in an unsupervised fashion [18, 29]. In this work, for the first time, we explore the solutions to leverage the self-supervised knowledge by distillation rather than simple fine-tuning as adopted in most existing works in SSL. For this aim, we establish a new benchmark using various self-supervised teacher networks proposed in the recently advanced CLIP model [29], and propose an effective distillation scheme that generalizes well to distill the knowledge from different types of self-supervised teacher networks.

3 Distillation by Embedding Graph Alignment (EGA)

Our goal is to distill the knowledge learned from a self-supervised pre-trained teacher model to enhance the visual representation learning of a supervised student model on a visual recognition task. As self-supervised learners are often trained to discriminate individual image instances [7, 9, 15, 55], the fine-grained knowledge encoded in a self-supervised teacher network can be distilled by learning to mimic the instance-instance relationships in the feature
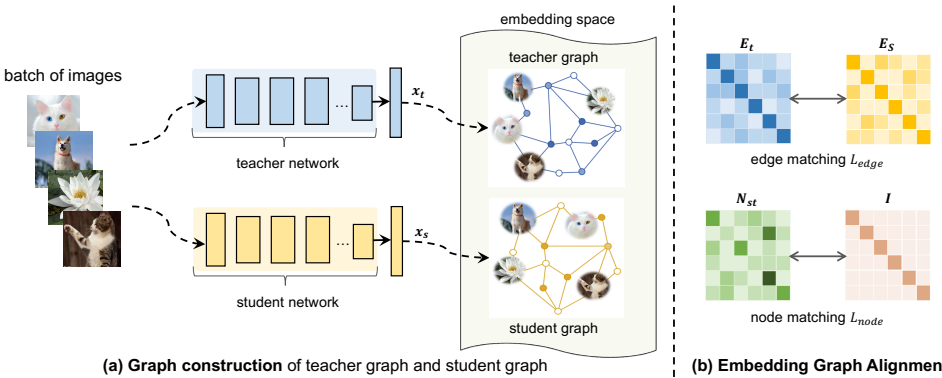
Fig. 2: To distill the knowledge from the self-supervised teacher network, we first construct the teacher graph and the student graph based on a batch of embeddings to capture the instance-instance correlation structure in the feature embedding space (Section 3.1). To transfer the structural correlation information learned by the teacher network to the student network, we align the teacher graph and the student graph by jointly optimizing an edge matching constraint and a node matching constraint (Section 3.2).

embedding space. Figure 2 presents the model overview of our proposed Embedding Graph Alignment (EGA) approach for distilling the knowledge from a self-supervised teacher network to a supervised student network. Given a batch of B image samples $\{I_1, I_2, ..., I_B\}$, we first feed the input samples to the teacher network and the student network respectively, resulting in a batch of teacher feature embeddings $\{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, ..., \mathbf{x}_{t_B}\}$ and a batch of student feature embeddings $\{\mathbf{x}_{s_1}, \mathbf{x}_{s_2}, ..., \mathbf{x}_{s_B}\}$. To represent the geometry and correlations among different feature embeddings, we construct a graph $\mathcal{G}$ in the embedding space to encode the instance-instance relationships (Section 3.1). For distilling the knowledge, we align the teacher graph and the student graph by jointly optimizing an edge matching constraint and a node matching constraint (Section 3.2), which work synergistically to align the features between teacher and student networks, and transfer the instance-instance correlations from the teacher to the student.

3.1 Graph Construction

To capture the fine-grained structural information in the embedding space, we construct a graph $\mathcal{G} = (X, E)$ to encode the structural correlations among a batch of embeddings, where X refers to a set of nodes (also known as vertices), and E represents a set of edges that encode the structural correlations among different instances: $E \subseteq \{(\mathbf{x}_i, \mathbf{x}_j) \mid (\mathbf{x}_i, \mathbf{x}_j) \in X \text{ and } \mathbf{x}_i \neq \mathbf{x}_j\}$. In the following, we detail how we compute the node embedding and the edge to construct the graph $\mathcal{G}$ in the embedding space.

Node. To learn the node embedding, we feed the extracted features from the teacher and student networks to individual node embedding layers (which are parameterized as linear projection layers). This also allows the different teacher and student networks to be deployed in our framework without checking which network needs an extra projection layer. Let $\mathbf{f}_t \in \mathbb{R}^{D_t}$ and $\mathbf{f}_s \in \mathbb{R}^{D_s}$ denote the feature vectors extracted directly from the teacher and student networks. We pass $\mathbf{f}_t$ and $\mathbf{f}_s$ to the two node embedding layers to obtain the projected embeddings $\mathbf{x}_t \in \mathbb{R}^D$ and $\mathbf{x}_s \in \mathbb{R}^D$, where D denotes the dimensionality of the embedding space shared between the teacher and the student models. The embeddings $\mathbf{x}_t, \mathbf{x}_s$ are utilized as the node representations in the teacher graph $\mathcal{G}_t$ and the student graph $\mathcal{G}_s$, which capture

the high-level semantic information of each input image. To distill knowledge from the teacher to the student model, an intuitive solution is the align the node embeddings $\mathbf{x}_t$, $\mathbf{x}_s$ directly through certain learning constraints, which we will detail in Section 3.2.

Edge. To capture the structural correlations among different instances, we propose to derive each edge in the graph to encode the correlation between every pair of images among the same batch based on the Pearson's correlation coefficient (PPC) – also known as the Pearson product-moment correlation coefficient (PPMCC). Formally, given n pairs of data points denoted as $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^1$, the correlation coefficient is defined as:

$$r_{xy} \overset{\text{def}}{=} \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

where n is the sample size, x_i, y_i are the individual data points, $\bar{x}, \bar{y}$ are the mean, s_x and s_y are the standard deviation. We apply the above formula to quantify the correlation between every pair of node embedding, and construct the edge based on this correlation. Specifically, given a pair of node embeddings denoted as $\mathbf{x}$ and $\mathbf{y}$ (where $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{y} \in \mathbb{R}^D$), we can compute their connected edge in a graph based on Eq. 1, as defined below.

$$e_{x,y} = \frac{\sum_{i=1}^{D}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{y}_i - \bar{\mathbf{y}})}{\sqrt{\sum_{i=1}^{D}(\mathbf{x}_i - \bar{\mathbf{x}})^2}\sqrt{\sum_{i=1}^{D}(\mathbf{y}_i - \bar{\mathbf{y}})^2}} \tag{2}$$

where D is the feature dimension. $e_{x,y}$ is the edge that connects the two node embeddings $\mathbf{x}$, $\mathbf{y}$, and quantifies their relationship through their correlation. Based on Eq. 2, one can build the teacher graph for a batch of node embeddings $\{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_B}\}$ by computing the edge between every pairs of node embeddings, while the same procedure can be applied for building the student graph. The constructed graph, therefore, encodes the pairwise correlations among all the instances in the same batch. We will elaborate on how to exploit this structural information in the following.

3.2 Embedding Graph Alignment

As motivated above, to distill the knowledge from a self-supervised teacher network, we aim to enforce a student network to mimic the fine-grained instance-wise correlations that are typically learned by a self-supervised learner. We encode these structural correlations by constructing the graph in the embedding space, and propose to distill such structural information by aligning the teacher graph and the student graph: named as Embedding Graph Alignment (**EGA**). Below, we first detail how we construct the edge matrix and node matrix, and describe how we formulate our distillation objectives to match the edges and nodes.

Edge matrix. The edge matrix is formulated based on Eq. 2, which encodes all the pairwise correlations among a batch of node embeddings $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_B\}$. Let $e_{i,j}$ denote the correlation between the embeddings $\mathbf{x}_i$ and $\mathbf{x}_j$ computed with Eq. 2. We can formulate an edge matrix as:

$$E(X, X) = (e_{ij}) \in \mathbb{R}^{B \times B} \tag{3}$$

where B is the batch size. All the diagonal elements on E are 1, given that its diagonal represents the self-correlations of each individual embedding. With Eq. 3, we can compute the edge matrices for both the teacher model and the student model using their batch of node embeddings: $X_t = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_B}\}$ and $X_s = \{\mathbf{x}_{s_1}, \mathbf{x}_{s_2}, \ldots, \mathbf{x}_{s_B}\}$. Accordingly, we can write

their corresponding edge matrices as $E_t = E(X_t, X_t)$ and $E_s = E(X_s, X_s)$, which encodes all the edges in two graphs.

Edge matching. To distill the structural correlation information captured in the teacher graph, we propose to align the edge matrices of the teacher graph and the student graph. Formally, we define an edge matching loss to match the edge information between two graphs, as written below.

$$\mathcal{L}_{edge} \triangleq \|E_t - E_s\|_2 \tag{4}$$

where $\mathcal{L}_{edge}$ explicitly enforces the student network to distill the same structural correlations learned by the teacher network by aligning the edge matrices E_t and E_s. Once trained with the above constraint, the student network is expected to obtain similar relative relations among different instances in the embedding space.

Node matrix. Although Eq. 4 helps to transfer the structural relations from the teacher to the student network, it does not explicitly align the individual embeddings between the teacher and the student networks. To encourage the student to mimic the representations learned by the teacher network, we propose to align their node embeddings of the same input image. Specifically, we define a node matrix where each element captures the correlation between the teacher and the student embeddings. We derive this matrix in a similar manner as our formula in Eq. 3:

$$N_{st} = E(X_t, X_s) \tag{5}$$

Conceptually, the node matrix N_{st} connects the teacher and student embeddings by building an edge among every pair of teacher and student embeddings to quantify the cross-correlations between the teacher and student models, which differs from the edge matrices that quantify the self-correlations among embeddings from the same network.

Node matching. To ensure the embeddings are aligned across the teacher and student networks, we impose the following node matching loss to ensure that the correlation between the teacher and student embeddings of the same input image is high (i.e. close to 1); while the correlation between different input images is low (i.e. close to 0). For this aim, we enforce the node matrix from Eq. 5 to align with an identity matrix $\mathcal{I}$, as defined below.

$$\mathcal{L}_{node} \triangleq \|N_{st} - \mathcal{I}\|_2 \tag{6}$$

where the node matrix N_{st} is encouraged to have its diagonal elements as 1 and the rest as 0, given that the diagonal encodes the correlation between the teacher and student embeddings of the same input image, and should yield high correlation values close to 1.

Embedding Graph Alignment. To align the teacher and the student graphs, we impose the embedding graph alignment loss for distillation, which contains two loss terms: the edge matching loss (Eq. 4) and the node matching loss (Eq. 6), as defined below.

$$\mathcal{L}_{EGA} = \mathcal{L}_{node} + \lambda \mathcal{L}_{edge} \tag{7}$$

where λ is a hyperparameter to balance two loss terms.

3.3 Model Optimization

Overall objective. Our learning objective includes a standard task objective and an embedding graph alignment loss λ_{EGA} for knowledge distillation. We consider image classification

in this work and the task objective is a standard cross-entropy loss $\mathcal{L}_{CE}$. The overall objective can be written as:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_{EGA}\mathcal{L}_{EGA} \tag{8}$$

Given that the teacher network is a self-supervised pre-trained model and it is not optimized for the downstream target task, we consider two standard distillation strategies to train the teacher and student networks: (1) mutual learning [40], which optimizes the teacher with the target objective $\mathcal{L}_{ce}$ and the student with the full objective (Eq. 8) *simultaneously*; (2) sequential learning [16], which optimizes the teacher and student networks *sequentially* by first training the teacher with the task objective $\mathcal{L}_{ce}$ and then training the student with the full objective (Eq. 8). For both strategies, the teacher's backbone is frozen and only its new added layers are trained. In our experiments, we find the two strategies perform similarly. We provide algorithm overviews on these strategies in supplementary.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on three image classification datasets. **CIFAR100** consists of 60,000 32×32 colour images, with 50,000 images for training and 10,000 images for testing. It has 100 classes and each class contains 600 images. **STL-10** contains a training set of 5,000 labeled images from 10 classes and 100,000 unlabeled images, and a test set of 8K images. It has 10 classes and the image size is 96x96. We only use the labeled data for training. **TinyImageNet** contains 100,000 colored images of 200 classes, and each class contains 500/50 training/validation images from ImageNet. The image sizes are downsized to 64x64.

Implementation details. We use PyTorch for our experiments. We test different types of student networks, including resnet8x4, shuffleNetV1, and VGG13. All the extracted image features from the student network are passed through a node embedding layer (Section 3.1), which maps each feature vector to a 256-D embedding by a linear projection. We also test different types of teacher networks, including self-supervised models and supervised models. Our self-supervised models include ViT-B/32, ViT-B/16 and RN101 [29]; while our supervised models include RN50, RN101 and WRN-40. Similarly, all the features from the teacher network are passed through a node embedding layer to get a 256-D embedding. For data augmentation, we apply standard strategies such as random cropping, random horizontal flipping, and normalization. For the hyperparameters, we set λ to 0.3 and λ_{EGA} as 0.8. For model training, we adopt the SGD optimizer, with an initial learning rate of 0.05, and decayed by 0.1 every 30 epochs after the first 150 epochs. The model is trained for 240 epochs with a batch size of 64. For distillation, we consider two training strategies: mutual learning and sequential learning as mentioned in 3.3.

4.2 Comparing to the State-of-the-Art

Compared methods. We compare our method to multiple representative state-of-the-art distillation methods. To ensure fair comparisons, we adopt the same network architectures, the same optimization scheme and the training strategies for all the methods, while using the customized distillation objectives for different methods. we compare against the following models: **KD** [16]: the first distillation method, using soft targets from the teacher to guide the

Method	Same student different teacher			Same teacher different student		
	ViT-B/32	ViT-B/16	RN101	Resnet8x4	ShuffleNetV1	VGG13
KD[■]	71.55	71.99	64.77	71.55	72.90	75.20
FitNet[■]	73.93	74.13	74.14	73.93	nan	75.56
PKT [■]	73.86	73.55	72.21	73.86	75.31	75.55
RKD[■]	73.34	73.42	73.7	73.34	73.93	76.41
NCE [■]	74.30	74.41	73.69	74.30	73.99	76.42
IRG [■]	75.11	74.72	74.17	75.11	74.79	75.98
CRD[■]	75.73	75.68	75.13	75.73	75.54	76.83
CCL[■]	75.91	76.13	75.08	75.91	76.14	**77.68**
EGA	**76.65**	**76.30**	**75.41**	**76.65**	**76.24**	77.59

Table 1: The teacher and student are trained ***simultaneously*** [■]. Best results are **bold**.

Method	Same student different teacher			Same teacher different student		
	ViT-B/32	ViT-B/16	RN101	Resnet8x4	ShuffleNetV1	VGG13
RKD[■]	73.36	72.43	73.92	73.36	72.62	73.26
CRD[■]	75.51	73.38	74.85	75.51	74.87	77.41
CCL[■]	75.98	39.56	74.22	75.98	76.05	77.54
EGA	**76.11**	**74.02**	**75.22**	**76.11**	**76.74**	**77.76**

Table 2: The teacher and student are trained ***sequentially*** [■]. Best results are **bold**.

student by KL divergence. **FitNet** [■]: a popular technique that aligns the features between the teacher and student networks by regression. **PKT** [■]: a probabilistic distillation model that matches the probabilistic distributions between the student and teacher by a divergence metric. **RKD** [■]: a relational distillation technique that transfers the distance-wise and angle-wise relations among features from the teacher to student model. **IRG** [■]: a distillation approach that models feature space transformation across layers and minimizes its MSE loss. **CRD** [■]: a contrastive distillation method that transfers knowledge by aligning the teacher and student's representations through instance-wise contrastive learning. **CCL** [■]: a recent distillation method to transfer knowledge across heterogeneous networks using a noise contrastive loss to align teacher and student's representations and a JSD loss to align the model predictions

Discussion. Among all the methods above, RKD and IRG are most related to our model as these are motivated to transfer the relational knowledge learned from the teacher to the student. However, our method EGA differs from other relational distillation approaches in multiple aspects: **(1)** We're the first to explore distillation from self-supervised teacher models and establish a new benchmark (Table1&2), while showing better performance on standard distillation using supervised teacher models (Table3). **(2)** We formulate graphs with node embedding layers, Pearson's correlation coefficients (PPC, Eq.1,2), and match graphs by enforcing cross-graph node matrix to be identity and aligning edge matrices with Frobenius norm (Eq.3-6), which greatly differs from these two methods: RKD computes relations as distances between pairs, angles among triplet and aligns relations; IRG models feature space transformation across layers and minimizes its MSE loss. **(3)** Our method shows more simplicity in implementation and achieves superior performance.

Evaluation on different network architectures. To evaluate how our model performs using different combinations of teacher and student networks, we deploy distillation using (1) the same student and different teacher, and (2) the same teacher and different student.

Table 1 shows our results on CIFAR100 with various teacher-student combinations. As shown, our EGA offers superior model performance in four out of five cases as compared to the best competitor CCL. When using the teacher and student as "ViT-B/32 + resnet8x4", "ViT-B/16 + resnet8x4", "RN101 + resnet8x4", and "ViT-B/32 + shuffleNetV1", EGA ob-

Method	Same student different teacher		
	RN101	RN50	WRN-40
KD[16]	74.69	74.82	74.77
FitNet[30]	58.73	76.27	75.58
PKT[22]	74.44	75.69	75.30
RKD [26]	72.45	72.25	72.48
NCE[10]	73.62	74.35	72.90
CRD [6]	75.52	75.50	75.84
CCL[10]	75.56	75.53	75.33
EGA	**75.77**	**76.36**	**75.97**

Table 3: Top 1 accuracy(%) of student network on CIFAR100. The teacher and student are trained ***simultaneously*** [40].

Method	CIFAR100	STL-10	TinyImageNet
KD [16]	71.55	84.35	54.68
FitNet [30]	76.04	84.15	59.97
PKT [22]	72.51	82.37	58.34
RKD [26]	73.34	83.13	58.15
NCE [10]	74.30	83.96	58.93
CRD [6]	75.73	82.40	60.34
CCL [10]	75.91	84.01	60.84
EGA	**76.65**	84.15	60.61
EGA + KD	76.49	**84.36**	**61.24**

Table 4: Top 1 accuracy (%) of student networks on CIFAR100, STL10, TinyImageNet. Teacher: self-supervised ViT-32 [29]. Student: resnet8x4. The teacher and student are trained ***simultaneously*** [40].

tains an accuracy of 76.65%, 76.30%, 75.41%, 76.24%, which are much higher than 75.91%, 76.13%, 75.08%, 76.14% by CCL. When using the teacher and student as and "ViT-B/32 + VGG13", EGA achieves very similar performance as CCL, obtaining an accuracy of 77.59% vs 77.68% by CCL which combines NCE and JSD for distillation.

In Table 2, we evaluate the sequential learning strategy of teacher and student in distillation. When comparing EGA to the top competitors, we find that EGA obtains the best overall performance among different combinations of teacher and student networks. In particular, EGA surpasses the best competitor CCL in five out of five cases, obtaining an accuracy of 76.11%, 74.02%, 75.22%, 76.74%, 77.76% when using the teacher and student as "ViT-B/32+resnet8x4", "ViT-B/16+resnet8x4", "RN101+resnet8x4", "ViT-B/32+shuffleNetV1", and "ViT-B/32+VGG13". Overall, our results in Table 1 and Table 2 indicate that EGA achieves strong model generalization when using different combinations of teacher and student networks, under two different training strategies for distillation. These results collectively show that EGA is capable of distilling self-supervised knowledge in an effective way.

Evaluation on distilling supervised knowledge. We focus on distilling self-supervised knowledge in this work, given that self-supervised models are cost-effective – they require no labels but learn good features. However, our model can also generalize to distilling knowledge from self-supervised networks. We evaluate this aspect in Table 3. As shown, our model EGA achieves the best model performance among all the competitors in all the comparisons, offering an accuracy of 75.77%, 76.36%, 75.97% vs 75.56%, 75.53%, 75.33% by the competitor CCL when using RN101, RN50, WRN-40 as the supervised teacher network. These results indicate that our EGA also works well in the scenario where the teacher networks are trained in a supervised manner.

Evaluation on CIFAR100, STL-10 and TinyImageNet. Table 4 shows our results on three image classification datasets, which transfer the self-supervised knowledge learned from the CLIP model [29] to a lightweight student network by training the two networks simultaneously. As Table 4 shows, our model GM obtains the overall superior model performance compared to other strong competitors. When comparing EGA+KD with the recent best competitor CCL which combines two losses (NCE+JSD), we find that EGA+KD outperforms CCL substantially, obtaining an accuracy of 76.49%, 84.36%, 61.24% on the three datasets vs 75.91%, 84.01%, 60.84% by CCL.

In Table 5, we evaluate another sequential learning strategy on the three datasets. By sequential learning, we first pre-train the teacher model on the downstream task and keep its weights frozen during distillation. As can be seen, when comparing EGA to multiple
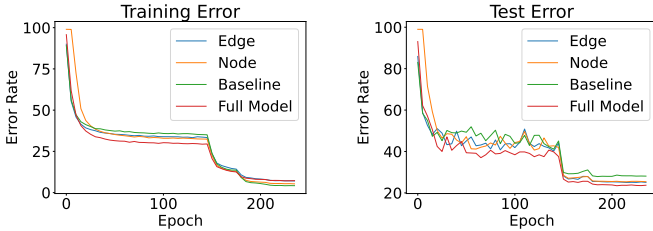
Fig. 3: The learning curves of baseline, EGA w edge matching loss only, EGA w node matching loss only, EGA on CIFAR100.

top competitors, the superiority of EGA remains the same. Overall, our results in Table 4 and Table 5 show that our model EGA works well for distilling self-supervised knowledge to boost the classification performance on various downstream image datasets. Moreover, EGA also works well under different distillation strategies, i.e., training the teacher and student simultaneously or sequentially. These results show the superior generalizability of EGA for distilling self-supervised knowledge in different scenarios.

Analyzing the learning dynamics. Figure 3 shows the learning curves of different models on CIFAR100. We compare the baseline model (only $\mathcal{L}_{ce}$), edge matching loss model ($\mathcal{L}_{ce} + \lambda_{edge}\mathcal{L}_{edge}$), node matching loss model ($\mathcal{L}_{ce} + \lambda_{node}\mathcal{L}_{node}$) and our full EGA model. It is interesting to observe that our EGA is less overfitting on the training set compared to other models, especially the baseline. We find that EGA consistently performs best during training on the test set, obtaining the lowest error rate. Besides, when adding either $\mathcal{L}_{edge}$ or $\mathcal{L}_{node}$ with the $\mathcal{L}_{ce}$, the model outperforms the

Method	CIFAR 100	STL-10	TinyImageNet
RKD [24]	73.36	82.67	58.32
CRD [5]	75.51	78.76	60.82
CCL[10]	75.98	80.41	61.24
EGA	**76.11**	**83.01**	**61.85**

Table 5: Top 1 accuracy (%) of student networks on CIFAR100, STL10, TinyImageNet. Teacher: self-supervised ViT-32 [29]. Student: resnet8x4. The teacher and student are trained ***sequentially*** [16]. Best results are highlighted in **bold**.

baseline, which indicates that these two loss terms are indeed effective. Figure 3 again proves that our full model formulation EGA performs and generalizes much better than the EGA variants with a single loss term.

5   Conclusion

In this work, we introduced a new Embedding Graph Alignment (EGA) method for distilling self-supervised knowledge. To capture the fine-grained instance-wise correlations learned by the teacher network, we construct graphs to encode the structural information in the embedding space, and perform knowledge distillation by aligning the teacher graph and the student graph. We evaluated our EGA extensively on multiple image classification datasets in comparison to a wide variety of knowledge distillation methods. We show that EGA generalizes well on different datasets, under different combinations of teacher and student networks, and performs robustly when using different training strategies for distillation. Our results collectively demonstrate the superiority of EGA as a generic knowledge distillation technique.

References

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Asr is all you need: Cross-modal distillation for lip reading. In *ICASSP*, 2020.

[2] Samuel Albanie, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Emotion recognition in speech using cross-modal transfer in the wild. In *ACM MM*, 2018.

[3] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeuRIPS*, 2016.

[4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NeuRIPS*, 2014.

[5] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeuRIPS*, 2019.

[6] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *ACM SIGKDD*, 2006.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *NeuRIPS*, 2020.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[10] Yanbei Chen, Yongqin Xian, A Koepke, Ying Shan, and Zeynep Akata. Distilling audio-visual knowledge by compositional contrastive learning. In *CVPR*, 2021.

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

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

[13] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016.

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

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

[17] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[18] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[19] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017.

[20] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *CVPR*, 2019.

[21] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

[22] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[23] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.

[24] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *CVPR*, 2018.

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[26] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019.

[27] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *ECCV*, 2018.

[28] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *CVPR*, 2020.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[30] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[31] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Visual permutation learning. *IEEE TPAMI*, 2018.

[32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2019.

[33] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *NeuRIPS*, 2020.

[34] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, 2019.

[35] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[36] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[37] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021.

[38] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[39] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[40] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018.