Do Topological Characteristics Help in Knowledge Distillation?

Jungeun Kim [*1] **Junwon You** [*2] **Dongjin Lee** [*3] **Ha Young Kim** [1 4] **Jae-Hun Jung** [2 3]

Abstract

Knowledge distillation (KD) aims to transfer knowledge from larger (teacher) to smaller (student) networks. Previous studies focus on point-to-point or pairwise relationships in embedding features as knowledge and struggle to efficiently transfer relationships of complex latent spaces. To tackle this issue, we propose a novel KD method called TopKD, which considers the global topology of the latent spaces. We define *global topology knowledge* using the persistence diagram (PD) that captures comprehensive geometric structures such as shape of distribution, multiscale structure and connectivity, and the *topology distillation loss* for teaching this knowledge. To make the PD transferable within reasonable computational time, we employ approximated persistence images of PDs. Through experiments, we support the benefits of using global topology as knowledge and demonstrate the potential of TopKD. Code is available at https://github.com/jekim5418/TopKD

1. Introduction

Large-scale deep learning models with numerous training parameters have recently demonstrated outstanding performance. However, the massive computational demands pose limitations for on-device applications (Chen et al., 2021b;c). Thus, model compression has become an active research field. A primary approach in this field, knowledge distillation (KD), aims to improve the performance of a smaller network, referred to as a *student*, by transferring knowledge acquired from a well-trained larger network, known as a *teacher* (Hinton et al., 2015).

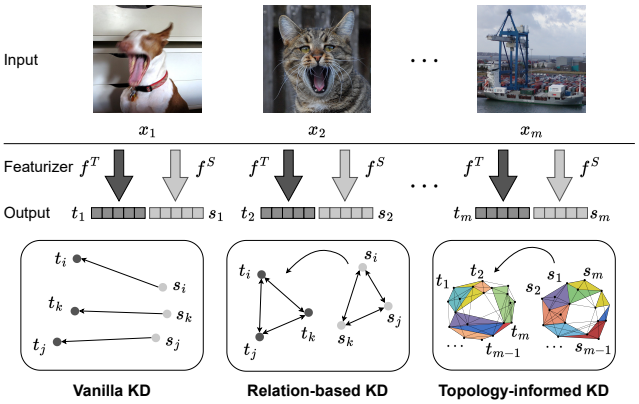


Figure 1: Topology-informed knowledge distillation. The proposed approach transfers the global topology of the entire embedding features from the teacher to the student model.

The most critical part of KD is determining which teacher knowledge to transfer. In the initial study, vanilla KD (Hinton et al., 2015) uses soft logits of the teacher network as the knowledge. To better imitate the teacher network, feature-based KD methods were introduced to mimic the internal feature representations. For instance, FitNet (Romero et al., 2015) improved performance by matching the feature values of the intermediate layers between the teacher and student networks in a point-to-point manner using the L_2 loss. This leads to excessively large loss values, making effective guidance challenging.

Meanwhile, relation-based KD methods aim to identify intricate relationships across embedding features (Peng et al., 2019; Park et al., 2019; Huang et al., 2022). Determining these relationships mainly involves defining similarity measures between distinct embedding features (Peng et al., 2019; Huang et al., 2022; Yang et al., 2022a) or assessing the structure between two or three pairs of embedding features within the latent space (Yim et al., 2017; Park et al., 2019). RKD (Park et al., 2019) defined the structure of the embedding features based on the distance and angle. These methods teach the model the full relationship via interactions between a few embedding features; however, relying on partial information to understand the entire structure is limited. Therefore, transferable knowledge that teaches the broader context of relationships between all embedding

[*]Equal contribution [1]Department of AI, Yonsei University, Seoul, South Korea [2]Department of Mathematics, POSTECH, Pohang, South Korea [3]Graduate School of AI, POSTECH, Pohang, South Korea [4]Graduate School of Information, Yonsei University, Seoul, South Korea. Correspondence to: Ha Young Kim <hayoung.kim@yonsei.ac.kr>, Jae-Hun Jung <jung153@postech.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

features should be defined. However, it is difficult to consider all arbitrarily multiple relationships of the features simultaneously. To address this problem, we consider a topology-based method.

Persistent homology (PH), a primary method in topological data analysis (TDA), provides an efficient method of calculating the global topology of data. In addition, PH systematically analyzes homological variations across multiple scales and discerns their persistence over different resolutions. This process unveils comprehensive structural information regarding the given data, including shape of distribution, multiscale structure, and connectivity. Moreover, PH visualizes such information in a persistence diagram (PD) that summarizes the birth and death of the topological features with the resolution. Despite these advantages, PH has only been employed as a preprocessing step for raw data in KD (Du et al., 2022; Jeon et al., 2024). However, if PH is integrated directly into the intermediate layers, it could offer valuable insight into the structure of the embedding features in the latent space.

In this context, we propose topology-informed KD (TopKD), a novel approach that leverages the global topology knowledge, comprising the comprehensive and entire structure of all embedding features (Fig. 1). For TopKD, we define global topology knowledge using the PD and the topology distillation loss for teaching this knowledge. However, directly computing PDs at each learning iteration is impractical or impossible with reasonable computational complexity. To render it possible, we replace PDs with persistence images (PIs) (Adams et al., 2017b) approximated using RipsNet (de Surrel et al., 2022). That is, TopKD matches the approximated PIs of the teacher and student networks with the topology distillation loss. To evaluate TopKD, we conduct extensive experiments on image classification with the CIFAR-100 and ImageNet-1K datasets. In addition, we provide ablation studies to explore TopKD, error analyses of approximated PIs, and topological visualization of results. The main contributions of our TopKD are as follows:

- We propose a topology-based KD method by defining the global topology knowledge as a PD that reflects the comprehensive and entire structure of all embedding features. To the best of our knowledge, this is the first study to use TDA in the latent space in KD.
- TopKD efficiently enables learning by replacing impractical and computationally demanding PDs with approximated PIs. We also present its validity through experiments and analyses. This can address the limitations of previous research in which PDs were used only in preprocessing.
- Through extensive experiments, we confirm that TopKD surpasses vanilla KD, thereby showing the effectiveness of global topology, and its potential through competitive performance with other KD methods.

2. Related Works

2.1. Knowledge Distillation

KD aims to enhance performance by transferring knowledge obtained from a pretrained high-performing large model to a smaller one (Hu et al., 2023). Numerous KD models can be categorized as logit-based (Hinton et al., 2015; Zhao et al., 2022; Huang et al., 2022), feature-based (Romero et al., 2015; Ahn et al., 2019; Heo et al., 2019; Yang et al., 2022b; Deng et al., 2022; Zong et al., 2022; Liu et al., 2023), or relation-based KD methods. The vanilla KD (Hinton et al., 2015) and FitNet (Romero et al., 2015) are representative logit-based and feature-based methods, respectively.

Rather than depending on fixed representations, relation-based KD focuses on relationships and interactions between embedding features (Zagoruyko & Komodakis, 2016; Passalis & Tefas, 2018; Tung & Mori, 2019; Peng et al., 2019; Yang et al., 2020; Li et al., 2020; Chen et al., 2021a;c; Wang et al., 2023). Contrastive learning (Tian et al., 2019; Zhu et al., 2021) and diverse similarity matrices, such as the Pearson correlation coefficient (Huang et al., 2022), activation similarity matrix (Tung & Mori, 2019), attention matrix (Zagoruyko & Komodakis, 2016; Guo et al., 2023), Gaussian radial basis function (Li et al., 2020), and cosine similarity (Park et al., 2019; Wang et al., 2023) have been used to quantify the relation between embedding features.

Furthermore, some studies have assessed the structural proximity between embedding features (Yim et al., 2017; Park et al., 2019; Liu et al., 2019). To measure the distance between embedding features, FSP (Yim et al., 2017) calculated the flow of solution procedure matrix to determine the direction of features. In addition, a study has defined distance by constructing an instance relation graph, where the feature values are represented as vertices, and the Euclidean distances between pairs are depicted as edges (Liu et al., 2019). The RKD (Park et al., 2019) proposed distance-wise and angle-wise distillation losses to measure the proximity of two or three pairs of embedding features. No previous studies have used topological characteristics to identify the overall structure of embedding features as knowledge. Instead, most have tried to teach the entire relationship through fragmented pairwise relationships of a single type. In contrast, TopKD identifies the comprehensive and overall structures, including shape of distribution, multiscale structure, and connectivity, by exploiting the topology of all embedding features in the latent space.

2.2. Persistent Homology in Deep Learning

Numerous efforts have been made to integrate topological information, particularly PH, into machine learning for ge-

ometric data analysis (Pachauri et al., 2011; Reininghaus et al., 2015; Anirudh et al., 2016; Som et al., 2018; Nawar et al., 2020; Barannikov et al., 2021b;a; Trofimov et al., 2022). Despite the solid theoretical background of PH, it is not straightforward to feed PD into deep neural networks (DNNs) because they are defined as multisets that could have varying sizes across samples. There have been several efforts to convert a PD into a fixed-size vector to address this problem, including the Betti sequence (Umeda, 2017), persistence landscape (Bubenik et al., 2015), and PIs (Adams et al., 2017a). Serving as alternative data representations or additional input modalities for DNNs, such vectorization methods have been applied across fields, including time-series analyses (Umeda, 2017), astrophysics (Bresten & Jung, 2019), and medical imaging (Hajij et al., 2021). Numerous previous studies have limited the numerical experiment to domain-specific datasets (Stucki et al., 2023; Hu et al., 2022) or small-scale datasets such as CIFAR-10 (Barannikov et al., 2021b; Purvine et al., 2023) and MNIST (Barannikov et al., 2021b; Trofimov et al., 2022; Davies et al., 2023; Von Rohrscheidt & Rieck, 2023). However, we evaluate TopKD on a widely used large-scale benchmark such as ImageNet-1K.

Despite active research and the advantages of PH, in KD, PH has only been used for extracting information to augment the input data to the network (Du et al., 2022; Peng et al., 2024; Jeon et al., 2024). In other words, PH is employed solely to extract additional information from the input. In contrast, this study takes a step further by directly leveraging the topology as knowledge.

3. Background of Persistent Homology

Topological characteristics, such as connected components, loops, void spaces, and the Betti numbers, represent consistent properties of space that persist through continuous transformations, providing insights into the structure, shape, connectivity, and overall distribution within datasets. Specifically, the connected components signify clusters of associated points, loops indicate the presence of boundaries or closed paths within a dataset, and void spaces depict regions or areas without points. The Betti numbers quantitatively encapsulate all these characteristics. PH is a method that analyzes the creation and destruction of these topological characteristics by exploring the homology of spaces across different scales. A PD is a visual tool that illustrates the results of PH (Dey & Wang, 2022).

Barcode and Persistence Diagrams. For a topological space $\mathcal{X}$, a real-valued function $f : \mathcal{X} \to \mathbb{R}$ is defined on the space $\mathcal{X}$. For $\alpha \in \mathbb{R}$, the *α-sublevel set* $\mathcal{X}_\alpha$ of $(\mathcal{X}, f)$ is defined as $\{x \in \mathcal{X} : f(x) \leq \alpha\}$. As α goes from $-\infty$ to ∞, $\mathcal{X}_\alpha$ starts with the empty set and ends with the entire space $\mathcal{X}$. This nested sequence of sublevel sets $\emptyset = \mathcal{X}_{\alpha_0=-\infty} \hookrightarrow \mathcal{X}_{\alpha_1} \hookrightarrow \mathcal{X}_{\alpha_2} \hookrightarrow \cdots \hookrightarrow \mathcal{X}_{\alpha_n} = \mathcal{X}$ is called the *filtration* $\mathcal{F}_f$ induced by f. We compute the homology of each sublevel set to observe how it changes across the filtration. For example, a new k-dimensional (dim) hole appears in $\mathcal{X}_{\alpha_b}$ and merges with $\mathcal{X}_{\alpha_d}$, where $\alpha_b \leq \alpha_d$. These values of α_b and α_d are referred to as the *birth time* and *death time*, respectively, of the k-dim homology, and this homology *persists* on the interval $[\alpha_b, \alpha_d]$ and $\alpha_d - \alpha_b$ called *persistence*. The family of these intervals is the *persistence barcode* of $(\mathcal{X}, f)$, a multiset of points in $\{(x, y) \in \overline{\mathbb{R}}^2 \mid x \leq y\}$, where $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. The PD visualizes the barcode, as illustrated in Fig. 6(b) in the appendix.

Filtrations for point clouds. For a point set P in a metric space (M, d) (e.g., $(\mathbb{R}^n, d_u)$, where d_u is the Euclidean distance), a real-valued function $f : M \to \mathbb{R}$ is defined as $v \mapsto \min_{x \in P} d(v, x)$. Then, the *$\alpha$-sublevel set* of (M, f) represents the union of closed balls $\overline{B_x(\alpha)}$ centered at $x \in P$ with a radius of α. Computing singular homology groups is cumbersome. In the case of point cloud data (PCD), the union of balls can be replaced with simplicial complexes such as the Čech, Vietoris-Rips (Rips), and alpha complexes, which are commonly used. Due to the computational costs of the Čech complex and the potential transformation of an empty interval (Maria et al., 2014) of the alpha complex, we chose the Rips complex $\mathbb{VR}^r(P)$, an approximation of the Čech complex, defined as $\mathbb{VR}^r(P) = \{\sigma = \{p_0, ..., p_k\} | d_u(p_i, p_j) \leq 2r \text{ for any } p_i, p_j \in \sigma\}$ for PCD P and $p_i \in P$. Then, the Rips filtration $\mathcal{R}(P)$ is given by $\{\mathbb{VR}^\alpha(P) \hookrightarrow \mathbb{VR}^{\alpha'}(P)\}_{\alpha \leq \alpha'}$. Such filtration enables us to examine multiscale relationships across all data by varying α, which is the distance threshold between two points.

Vectorization of the PD. Integrating the PD into DNNs requires vectorization. Among the various vectorization methods, we selected PIs due to the possibility of steering persistence, which is critical for global topology knowledge. PIs (Adams et al., 2017a) transform a barcode into a vector while reflecting the persistence and density of points in the barcode. First, a barcode B is rotated via the map $T : \mathbb{R}^2 \to \mathbb{R}^2$, $(b, d) \mapsto (b, d - b)$. Next, the persistence surface $\rho_B : \mathbb{R}^2 \to \mathbb{R}$ corresponding to B is defined as $\rho_B(z) = \sum_{u \in T(B)} w(u) g_u(z)$, where $w : \mathbb{R}^2 \to \mathbb{R}$ is a weight function controlling the effect of persistence (Fig. 6 (c) in the appendix) and $g_u : \mathbb{R}^2 \to \mathbb{R}$ is a Gaussian function defined as $g_u(z) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{||z-u||^2}{2\sigma^2}\right)$ with a mean u and variance σ^2. The weight function $w(x, y) = 10(\tanh(y) + \ln(100x + 1))$ is designed to increase the influence of points with long persistence and reflect the global structure by weighting points according to the birth time. Last, PIs are realized by discretizing the surface via integration ρ_B over each subdomain. We let $\sqcup_i P_i$ be the partition of a compact

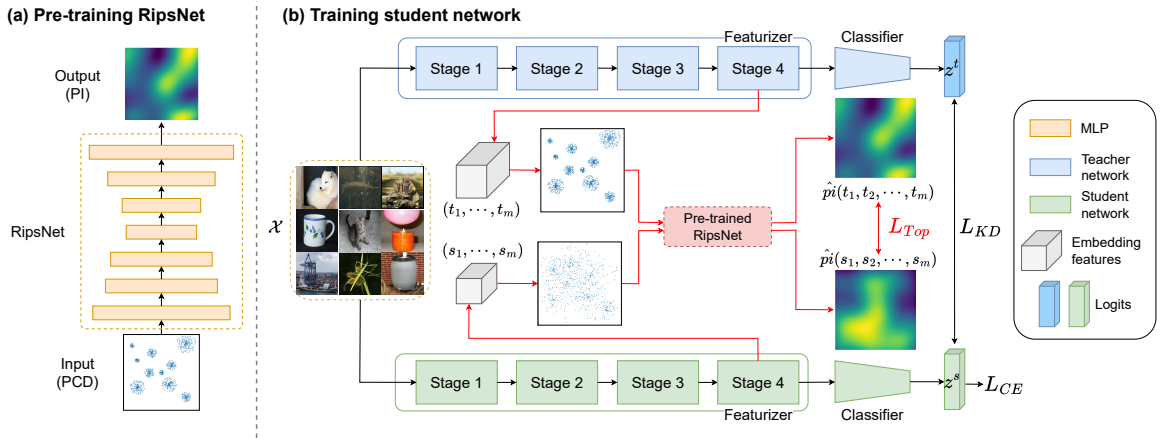


Figure 2: The overall training process for TopKD involving two stages: (a) training RipsNet and (b) training the student model. RipsNet is trained with the PCD and PIs generated from the featurizer output of the teacher network. The pretrained RipsNet is frozen and used to extract PIs from both networks to mimic the global topology. The student network is updated with $\mathcal{L}_{CE}$, $\mathcal{L}_{KD}$, and $\mathcal{L}_{Top}$.

subset $A \subseteq \mathbb{R}^2$ (in practice, a rectangular domain divided into $n \times n$ pixels). The PIs $(I(\rho_B)_{P_i})_i$ are the collection of pixels, where $I(\rho_B)_{P_i} = \iint_P \rho_B dz$.

4. Method

This section revisits conventional KD and introduces the proposed TopKD based on global topology knowledge, PI approximation, and topology distillation loss.

4.1. Preliminaries

Notation. Given pretrained teacher (T) and student (S) models, each model consists of a featurizer and classifier. For the teacher model, the output of any layer of the featurizer for a training sample x_i is represented as $f^T(x_i)$, and the preactivated output of the classifier, referred to as logits, is indicated as $z^T(x_i)$. Similarly, the outputs of the featurizer and classifier of the student network are expressed as $f^S(x_i)$ and $z^S(x_i)$, respectively. Furthermore, the softmax function is denoted as σ, and the temperature is represented as τ. If $\mathcal{X} = \{x_1, x_2, \cdots, x_N\}$ is a set of input samples where $\mathcal{Y} = \{y_1, y_2, \cdots, y_N\}$ is a set of true labels comprising C categories with N samples, the student model for image classification is trained by minimizing the cross-entropy (CE) loss, $\mathcal{L}_{CE} = \sum_{x_i \in \mathcal{X}} CE(\sigma(z^S(x_i)), y_i)$.

Conventional knowledge distillation. In vanilla KD (Hinton et al., 2015), the KD loss function, denoted as $\mathcal{L}_{KD}$, is a Kullback–Leibler (KL) divergence designed to minimize the soft logits of the teacher and student as training progresses:

$$\mathcal{L}_{KD} = \sum_{x_i \in \mathcal{X}} KL(\sigma(z^T(x_i)/\tau), \sigma(z^S(x_i)/\tau)). \quad (1)$$

Therefore, in conventional KD, the student model is trained with the final loss denoted as $\mathcal{L} = \alpha\mathcal{L}_{CE} + \beta\mathcal{L}_{KD}$, where the hyperparameters α and β control the significance of $\mathcal{L}_{CE}$ and $\mathcal{L}_{KD}$, respectively.

4.2. Topology-informed Knowledge Distillation

What knowledge should be distilled? As described in Section 3, the PH can offer a comprehensive and multifaceted explanation of the latent space of the teacher network. To transfer the PH of the embedding features, we define the PD of the embedding features as the global topology knowledge. These PDs provide topological summaries of the embedding features encompassing structural details about the similarity between data points, their overall distribution, and interactions and distances between them. For the effective integration of PDs, we use the PI as a vectorization, which can strengthen the global features by steering the influence of persistence.

Approximating PIs. Accurately calculating PDs and PIs for each batch requires heavy computational demands. Thus, TopKD approximates PIs using RipsNet to address this problem. While such approximation enables rapid calculation of PIs, it necessitates training specific to the task at hand. First, we created PCDs from the training data $\mathcal{X}$ using a pretrained teacher model T, as depicted in Fig. 2(a). We let $\mathcal{X}^m$ be a set of m-tuples of distinct samples in $\mathcal{X}$. In this case, m is the batch size. For $\mathcal{X}^m$, the set of PCDs P^T is generated as follows:

$$P^T = \{(f^T(x_1), .., f^T(x_m)) \mid (x_1, .., x_m) \in \mathcal{X}^m\}. \quad (2)$$

If the output size of the featurizer is $[h, w]$ with c channels, we consider $f^T(x_i)$ as a point in $\mathbb{R}^{h \times w \times c}$ or a point in $\mathbb{R}^c$ by passing through the pooling layer. We use the Gudhi

Table 1: Top-1 accuracy (%) comparison on CIFAR-100 with other KD approaches. Teacher and student networks have the same architectural style. Blue inverted triangles indicate lower performance than KD; red triangles signify better performance than KD. "-" indicates the absence of any available results. Relation denotes the relationships between a specific number of points (e.g., pairwise or triple-wise) to extract knowledge.

Distillation Mechanism	Knowledge	Relation	Teacher	ResNet56	ResNet110	ResNet110	ResNet32×4	WRN-40-2	WRN-40-2	VGG13
			Acc.	72.34	74.31	74.31	79.42	75.61	75.61	74.64
			Student	ResNet20	ResNet20	ResNet32	ResNet8×4	WRN-16-2	WRN-40-1	VGG8
			Acc.	69.06	69.06	71.14	72.50	73.26	71.98	70.36
Logit	Soft logits	-	KD	70.66	70.67	73.08	73.33	74.92	73.54	72.98
Feature	Feature value	-	FitNet	69.21 ▼	68.99 ▼	71.06 ▼	73.50 ▲	73.58 ▼	72.24 ▼	71.02 ▼
	Attention map	-	AT	70.55 ▼	70.22 ▼	72.31 ▼	73.44 ▲	74.08 ▼	72.77 ▼	71.43 ▼
	Variational distribution	-	VID	70.38 ▼	70.16 ▼	72.61 ▼	73.09 ▼	74.11 ▼	73.30 ▼	71.23 ▼
	Preactivation feature	-	OFD	70.98 ▲	-	73.23 ▲	74.95 ▲	75.24 ▲	74.33 ▲	73.95 ▲
Relation	Correlation coefficient	Pair	CC	69.63 ▼	69.48 ▼	71.48 ▼	72.97 ▼	73.56 ▼	72.21 ▼	70.71 ▼
	Similarity matrix	Pair	SP	69.67 ▼	70.04 ▼	72.69 ▼	72.94 ▼	73.83 ▼	72.43 ▼	72.68 ▼
	Direction	Pair	FSP	69.95 ▼	70.11 ▼	71.89 ▼	72.62 ▼	72.91 ▼	-	70.23 ▼
	Distance&angle	Pair/triple	RKD	69.61 ▼	69.25 ▼	71.82 ▼	71.90 ▼	73.35 ▼	72.22 ▼	71.48 ▼
	Probability of features	Pair	PKT	70.34 ▼	70.25 ▼	72.61 ▼	73.64 ▲	74.54 ▼	73.45 ▼	72.88 ▼
	Contrastive learning	Pair	CRD	71.16 ▲	71.46 ▲	73.48 ▲	75.51 ▲	75.48 ▲	74.14 ▲	73.94 ▲
	Contrastive learning	Pair	CRCD	73.21 ▲	72.33 ▲	74.98 ▲	76.42 ▲	76.67 ▲	75.95 ▲	74.97 ▲
Topology	Global topology	All	Ours	71.58 ▲	71.47 ▲	73.77 ▲	75.40 ▲	75.75 ▲	74.43 ▲	74.01 ▲

Table 2: Top-1 accuracy (%) comparison on CIFAR-100 with other KD approaches. These teacher and student networks have different architectural styles.

Distillation Mechanism	Knowledge	Relation	Teacher	VGG13	ResNet50	ResNet50	ResNet32×4	ResNet32×4	WRN-40-2
			Acc.	74.64	79.34	79.34	79.42	79.42	75.61
			Student	MobileNetV2	MobileNetV2	VGG8	ShuffleNetV1	ShuffleNetV2	ShuffleNetV1
			Acc.	64.60	64.60	70.36	70.50	71.82	70.50
Logit	Soft logits	-	KD	67.37	67.35	73.81	74.07	74.45	74.83
Feature	Feature value	-	FitNet	64.14 ▼	63.16 ▼	70.69 ▼	73.59 ▼	73.54 ▼	73.73 ▼
	Attention map	-	AT	59.40 ▼	58.58 ▼	71.84 ▼	71.73 ▼	72.73 ▼	73.32 ▼
	Variational distribution	-	VID	65.56 ▼	67.57 ▲	70.30 ▼	73.38 ▼	73.40 ▼	73.61 ▼
	Preactivation feature	-	OFD	69.48 ▲	69.04 ▲	-	75.98 ▲	76.82 ▲	75.85 ▲
Relation	Correlation coefficient	Pair	CC	64.86 ▼	65.43 ▼	70.25 ▼	71.14 ▼	71.29 ▼	71.38 ▼
	Similarity matrix	Pair	SP	66.30 ▼	68.08 ▲	73.34 ▼	73.48 ▼	74.56 ▲	74.52 ▼
	Distance&angle	Pair/triple	RKD	64.52 ▼	64.43 ▼	71.50 ▼	72.28 ▼	73.21 ▼	72.21 ▼
	Probability of features	Pair	PKT	67.13 ▼	66.52 ▼	73.01 ▼	74.10 ▲	74.69 ▲	73.89 ▼
	Contrastive learning	Pair	CRD	69.73 ▲	69.11 ▲	74.30 ▲	75.11 ▲	75.65 ▲	76.05 ▲
Topology	Global topology	All	Ours	68.83 ▲	69.12 ▲	74.25 ▲	75.04 ▲	76.33 ▲	76.18 ▲

library (Maria et al., 2014) to compute the PDs and PIs. The PDs are calculated by using the Rips complex. To compute PIs, we set the grid size to 20×20 and the weight function $w(x, y)$ to $10(\tanh(y) + \ln(100x + 1))$. For the other parameters, we refer to the experiments on RipsNet. The standard deviation of the Gaussian kernel is determined by the distance between the points of the PD, specifically the first five quantiles. The birth-death time ranges are set from the minimum to the maximum across all PDs. For the given PCD $(t_1, .., t_m)$ in P^T, the exact PI is represented as $pi(t_1, .., t_m)$, and the approximated PI is represented as $\hat{pi}(t_1, .., t_m)$. We approximate the PIs minimizing the $\mathcal{L}_2$ loss function defined as follows:

$$\mathcal{L}_{RN} = \sum_{(t_1,..,t_m)\in P^T} \mathcal{L}_2(\hat{pi}(t_1, .., t_m), pi(t_1, .., t_m)). \quad (3)$$

When the training is completed, RipsNet is frozen during the training of the student network to approximate PIs of the embedding features from the teacher and student networks, denoted as $\hat{pi}(t_1, .., t_m)$ and $\hat{pi}(s_1, .., s_m)$, respectively, where $t_i = f^T(x_i)$ and $s_i = f^S(x_i)$.

Topology distillation loss. To enable the student to mimic PIs of the teacher, we define the topology distillation loss function $\mathcal{L}_{Top}$ as follows:

$$\mathcal{L}_{Top} = \sum_{(x_1,..,x_m)\in\mathcal{X}^m} \mathcal{L}_2(\hat{pi}(t_1, .., t_m), \hat{pi}(s_1, .., s_m)). \quad (4)$$

The $\mathcal{L}_2$-norm was chosen because PIs are elements of $\mathbb{R}^n$, and the most natural metric in this context is the Euclidean distance. If the channel dimensions of t_i and s_i differ, a 1×1 convolution is applied to s_i. The logits represent high-level task-relevant knowledge; therefore, to use it further, we define the final loss function for TopKD by combining the vanilla KD loss with the topology distillation loss, as follows:

$$\mathcal{L}_{Total} = \alpha\mathcal{L}_{CE} + \beta\mathcal{L}_{KD} + \gamma\mathcal{L}_{Top} \quad (5)$$

where α, β, and γ are the hyperparameters that control the weights of $\mathcal{L}_{CE}$, $\mathcal{L}_{KD}$, and $\mathcal{L}_{Top}$, respectively. Fig. 2(b)

Table 3: Top-1 and top-5 accuracy (%) (Acc.) comparison on the ImageNet-1K validation dataset with ResNet34 as the teacher and ResNet18 as the student network. The best accuracy values are bolded, and "-" indicates the absence of any available results.

Acc.	Teacher	Student	AT	KD	SemCKD	OFD	CRD	CAT-KD	RKD	ReviewKD	DKD	SRRL	MGD	DistPro	NORM	Ours
Top-1	73.31	70.00	70.59	70.68	70.87	71.08	71.17	71.26	71.34	71.61	71.70	71.73	71.80	71.89	72.14	**73.60**
Top-5	91.42	89.60	89.73	90.16	-	-	90.13	90.45	90.37	**90.51**	90.41	-	90.40	-	-	90.50

Table 4: Top-1 and top-5 accuracy (%) on the ImageNet-1K validation dataset with ResNet50 as the teacher and MobileNetV2 as the student network.

Acc.	Teacher	Student	AT	KD	OFD	CRD	CAT-KD	RKD	ReviewKD	DKD	SRRL	MGD	DistPro	NORM	Ours
Top-1	76.16	66.20	69.56	68.58	71.25	71.37	72.24	71.32	72.56	72.05	72.49	72.59	73.26	74.26	**76.80**
Top-5	92.86	85.80	89.33	88.98	90.34	90.41	91.13	-	91.00	91.05	-	90.94	-	-	**92.80**

reveals that the student network is updated using a newly defined topology distillation loss, aiming to mimic the PIs of the teacher, and the existing vanilla KD loss.

5. Experiments

5.1. Datasets and Implementation Details

Baselines. Consistent with previous studies (Tian et al., 2019; Chen et al., 2021c; Guo et al., 2023), we compare TopKD with representative KD models for each distillation mechanism including KD (Hinton et al., 2015), FitNet (Romero et al., 2015), AT (Zagoruyko & Komodakis, 2016), SP (Tung & Mori, 2019), CC (Peng et al., 2019), VID (Ahn et al., 2019), RKD (Park et al., 2019), PKT (Passalis & Tefas, 2018), FSP (Yim et al., 2017), CRD (Tian et al., 2019), CRCD (Zhu et al., 2021), SemCKD (Chen et al., 2021a), OFD (Heo et al., 2019), DKD (Zhao et al., 2022), ReviewKD (Chen et al., 2021c), SRRL (Yang et al., 2021), MGD (Yang et al., 2022b), DistPro (Deng et al., 2022), DPK (Zong et al., 2022), NORM (Liu et al., 2023), and CAT-KD (Guo et al., 2023).

CIFAR-100 (Krizhevsky et al.) is a 32×32 pixel color image dataset, comprising 50K training and 10K test images, for a total of 60K images. It consists of 100 classes, each with 600 images, grouped into 20 superclasses, with each image annotated for a specific class and the corresponding superclass.

ImageNet-1K (Deng et al., 2009) is a large-scale image dataset consisting of 1K categories, 1.28M training images, and 50K validation images.

The training details of the student model and RipsNet are described in Appendix B and F.

5.2. Main Results

Results on CIFAR-100. Tables 1 and 2 present the accuracy for pairs when the teacher and student networks have the same and different structures on CIFAR-100, respectively. For a fair comparison, we followed the experimental settings for CRD, as in previous studies, and compared the current results with the values specified in paper (Chen et al., 2021c; Deng et al., 2022; Zong et al., 2022; Liu et al., 2023; Guo et al., 2023). TopKD surpasses both the student model and vanilla KD by a large margin, demonstrating the effectiveness of using the global topology as knowledge. In addition, TopKD displays competitive results compared to other representative models of existing KD mechanisms. These results highlight the potential of TopKD. Notably, TopKD, mimics the PD that includes information on the similarity, distance, and distribution between embedding features, outperforms CC, SP, RKD, PKT, and FSP, which mimic only fragmented information. The presented performance results from approximating the PI, and we anticipate that the performance will be further enhanced if the actual PIs or improved approximated PIs are used.

Results on ImageNet-1K. We evaluate the proposed method on the ImageNet-1K validation set to demonstrate its effectiveness on a large-scale dataset. Similar to the previous experimental setup, Tables 3 and 4 reveal top-1 and top-5 accuracy when the teacher and student structures are homogeneous and heterogeneous, respectively. Further, TopKD achieves the best results in top-1 accuracy compared to other baselines. Moreover, TopKD surpasses the top-1 performance of the teacher. The results imply that TopKD enhances the performance of the student model regardless of the dataset scale.

5.3. Ablation Study

In this section, to conduct ablation studies, we set ResNet56 as the teacher network and ResNet20 as the student network on CIFAR-100.

Dimension of PD. As described in Section 3, k-dim PDs exist based on the creation and disappearance of k-dim holes (e.g., connected components, loops, etc.) in the PCD. Each dim PD exhibits distinct topological characteristics. Therefore, we perform ablation experiments on the homology

dimension of PDs. As 0 and 1-dim PD are generally used due to the computational cost, we considered only those. Specifically, 0-dim PDs identify connected components, and 1-dim PDs detect loop structures. In Table 5, matching 0-dim PDs yields the best results. This observation can be attributed to the fact that 0-dim PDs contain information about clusters (Güzel & Kaygun, 2022), making it more beneficial for classification tasks. 0 and 1-dim PDs yield better results than 1-dim PDs but lower results than 0-dim PDs. This is because when generating PIs using both 0 and 1-dim PDs, the creation time of all 0-dim homology classes is 0. Therefore, unlike when generating only 0-dim PD (Fig. 4), the information of the 0-dim PD is encoded only on the left side of the PI (Fig. 6 in Appendix) as x-axis in the PI indicates the filtration value. Consequently, 0-dim characteristics occupy the small region of the PI, which impedes the distillation of information about clusters in the latent space.

Table 5: Ablation results on the homology dimension of persistent diagrams.

Dimension	0	1	0&1
Accuracy	71.58	71.22	71.36

Number of points in PCD. In general, with sufficient data, the underlying manifold within the data is approximated more accurately, resulting in less noisy PDs (Chazal et al., 2014). Thus, we conduct an ablation analysis on the number of points in PCD by varying batch size, doubling it from 8 to 256. As illustrated in Fig. 3, TopKD underperforms the student at batch size 8. This can be interpreted as insufficient data to adequately learn the global topology, thus leading to topology distillation loss interfering with the training. However, as the number of points exceeds 16, TopKD consistently outperforms the student network. This implies that there needs to be an adequate number of points to accurately discern the underlying all embedding features of topology, including the distribution and geometric characteristics.

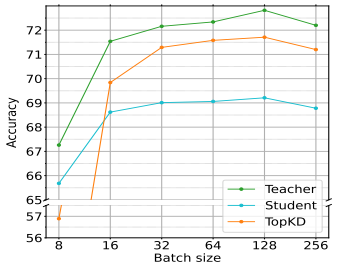


Figure 3: TopKD performance by batch size.

Effect of TopKD loss components. We assess the individual influence of each component comprising the total loss of TopKD through experiments. Table 6 summarizes the accuracy and variance results. Each row in the table indicates the performance with the gradual addition of KD loss and the proposed topology distillation loss to the classification loss. Adding $\mathcal{L}_{Top}$ to $\mathcal{L}_{CE}$ improves the performance compared to using $\mathcal{L}_{CE}$ alone, but this result is inferior to that of vanilla KD. However, using both $\mathcal{L}_{KD}$ and $\mathcal{L}_{Top}$ together offers better performance than using them separately and even surpasses it by a large margin. This outcome reveals that the topology distillation loss creates a benefical synergy with the KD loss.

Table 6: Performance comparison based on the presence or absence of elements comprising the TopKD loss (Eq. (5)). The coefficients of the loss are set to 1, 2, and 5, respectively.

$\mathcal{L}_{CE}$	$\mathcal{L}_{KD}$	$\mathcal{L}_{Top}$	Accuracy (Variance)
✓			69.06 (0.2275)
✓	✓		70.66 (0.2642)
✓		✓	69.72 (0.5939)
✓	✓	✓	71.58 (0.1885)

Optimal stages for TopKD. To identify the best intermediate layer features for TopKD, we evaluate the effect of matching the output features of each stage across four stages between the teacher and student models. Table 7 reveals that, when mimicking the PIs of the output features from the fourth stage of both networks, the model exhibits the best results. In addition, irrespective of which stage's features are matched, performance improved substantially compared to vanilla KD, suggesting that TopKD operates effectively without being sensitive to the level of features.

Table 7: Results of TopKD from various stages of teacher and student networks. The accuracy of the student model trained from scratch and vanilla KD is 69.1 and 70.66, respectively.

		Teacher stage			
		1	2	3	4
Student stage	1	71.27	71.51	71.48	71.36
	2	71.43	71.24	71.51	71.26
	3	71.38	71.05	71.07	71.34
	4	71.24	71.33	71.36	71.58

6. Analysis

6.1. Analysis Regarding Approximated PIs

$\mathcal{L}_{Top}$ as an upper bound of the exact loss. To verify the topology distillation loss, we demonstrate how matching the approximated PIs for the embedding features of the teacher and student networks affects the actual dis-

tance between their exact PIs. Here, we use the notation $pi_T = pi(t_1, \ldots, t_m)$ and $pi_S = pi(s_1, \ldots, s_m)$ for brevity. Similarly, we define $\hat{pi}_T = \hat{pi}(t_1, \ldots, t_m)$ and $\hat{pi}_S = \hat{pi}(s_1, \ldots, s_m)$. Then, the $\mathcal{L}_2$ distance between pi_T and pi_S has the following upper bound:

$$\|pi_T - pi_S\|_2 \leq \|pi_T - \hat{pi}_T\|_2 + \|\hat{pi}_T - \hat{pi}_S\|_2 + \|pi_S - \hat{pi}_S\|_2. \quad (6)$$

In the upper bound, the second term represents the topological distillation loss. The first and last terms on the right-hand side represent the approximation errors for the teacher and the student, respectively. Thus, the approximation capability is crucial for matching the exact PI of the student to that of the teacher network. The first term is typically small because RipsNet is trained on $(t_1, \ldots, t_m)$ with the loss function in Eq. (3). However, the last term is not directly minimized throughout the training process of the student.

Table 8: Approximation errors on CIFAR-100. $\mathcal{L}_{RN}$ denotes the training error for the teacher as in Eq. (3). The values are averaged across minibatches of the training dataset. The bolded value indicates the smallest error.

Teacher	Student	$\mathcal{L}_{RN}$ ($\|pi_T - \hat{pi}_T\|_2$)	$\|pi_S - \hat{pi}_S\|_2$		
			Student	Student w/ KD	Student w/ TopKD
VGG13	MobileNetV2	0.00229	0.02335	0.03408	**0.02103**
ResNet50	MobileNetV2	0.00218	0.00997	**0.00734**	0.00740
ResNet50	VGG8	0.00218	0.11178	0.01961	**0.01679**
ResNet32×4	ShuffleNetV1	0.00248	0.05301	0.05114	**0.04084**
ResNet32×4	ShuffleNetV2	0.00248	0.06427	0.07016	**0.00434**
WRN-40-2	ShuffleNetV1	0.00164	0.06591	0.05419	**0.05001**

Error analysis. We evaluate the approximation errors on the embedding features of the student and teacher networks using the CIFAR-100 dataset, and Table 8 presents the results. The experimental setup is detailed in Appendix G.1. The RipsNet approximates the exact PIs for the teacher network with errors of a magnitude around 2×10^{-3}, whereas the approximated PIs for the student networks exhibit larger errors. Our method has smaller errors compared to the students trained from scratch or with KD. This effectively tightens the upper bound in Eq. (6). Since RipsNet is trained on the embedding features of the teacher network, it produces more accurate approximations for embedding features with a similar structure to the teacher. Thus, we conjecture that the smaller errors observed in TopKD suggest that our method produces embedding features more closely aligned with the teacher network than other student networks. Fig. 4 presents the visualization of the exact and approximated PIs of 0-dim homology. Among the various methods, the teacher network achieves the most accurate approximations of PIs. We observe that TopKD produces more realistic PIs than others. On the other hand, the generated PIs across embedding features of student networks appear nearly identical. We speculate that this low variability in the approximated PIs results from the difference in the scale of embedding features between the teacher and student networks.

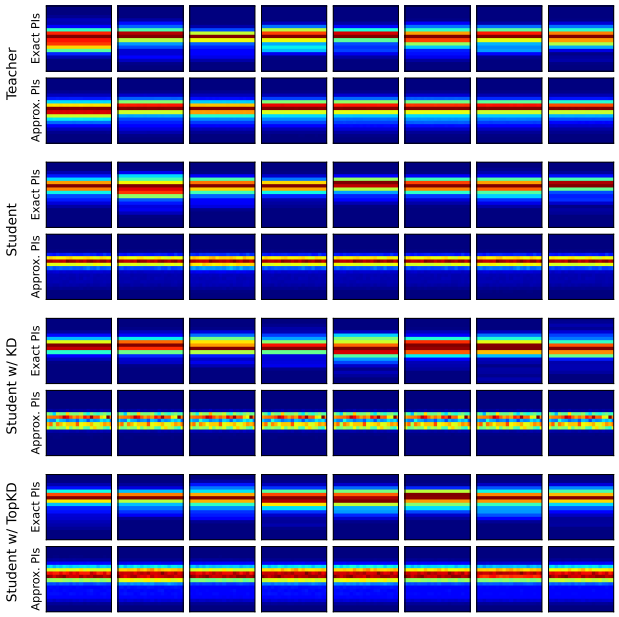


Figure 4: Exact and approximated PIs for the embedding features of minibatches extracted from the teacher (VGG13) and student (MobileNetV2) networks. Each column corresponds to a minibatch with a size of 64. The PIs exhibit straight-line shapes because the birth-times of 0-dim homology are zeros.

6.2. Visualization of Overall Topology

Fig. 5 visualizes the topological structure of the embedding features of the teacher (top left), student (top right), vanilla KD-trained student (bottom left), and TopKD-trained student (bottom right) through uniform manifold approximation and projection (UMAP) (McInnes et al., 2018) to present qualitative results. The teacher model displays a well-defined class clustering, suggesting an organized latent space conducive to strong performance. The vanilla KD and TopKD exhibit an enhanced clustering compared to the student network, yet these models still fall short of achieving the level of the teacher network. However, TopKD more effectively gathers points by class than vanilla KD, making clearer distinctions between classes, such as beaver, bee, aquarium fish, bear, and beetle. The quantitative analysis for the UMAP visualizations is presented in Appendix G.3.

7. Conclusion

In this paper, we proposed TopKD, a novel topological KD methodology based on PD, to teach comprehensive relationships of all embedding features. TopKD enables learning

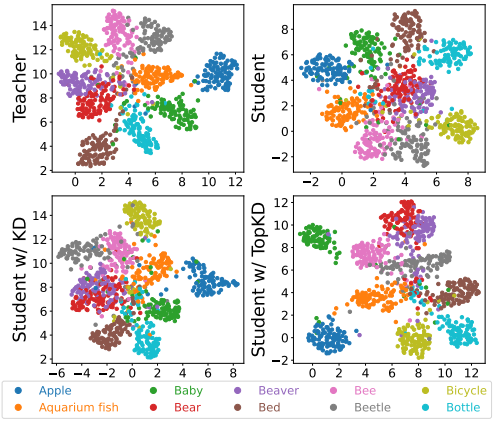


Figure 5: Visualization of embedding features of featurizer using UMAP, a dimension reduction technique based on topology and geometry. VGG13 is used for the teacher network, and MobileNetV2 for the student network. We extracted the first 10 out of 100 classes from the test data of CIFAR-100 for visualization.

PDs by replacing computationally demanding PDs with approximated PIs. Through experiments, we showed that utilizing global topology as knowledge is effective, achieving promising and competitive performance with baselines. Notably, TopKD successfully operated on the large-scale dataset ImageNet-1K and outperformed the teacher. To enhance TopKD in future work, we plan to integrate additional topological features such as Betti sequences and persistence landscapes. We believe that the TopKD methodology will serve as the cornerstone for research in distilling the topological characteristics of the latent space.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2023R1A2C200337911, RS-2023-00220762, 2021R1A2C3009648, NRF2021R1A6A1A1004294412, and RS-2023-00219980), and partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University) and No.2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017a. URL http://jmlr.org/papers/v18/16-337.html.

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017b.

Ahn, S., Hu, S. X., Damianou, A., Lawrence, N. D., and Dai, Z. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9163–9171, 2019.

Anirudh, R., Venkataraman, V., Natesan Ramamurthy, K., and Turaga, P. A riemannian framework for statistical analysis of topological persistence diagrams. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 68–76, 2016.

Barannikov, S., Trofimov, I., Balabin, N., and Burnaev, E. Representation topology divergence: A method for comparing neural network representations. *arXiv preprint arXiv:2201.00058*, 2021a.

Barannikov, S., Trofimov, I., Sotnikov, G., Trimbach, E., Korotin, A., Filippov, A., and Burnaev, E. Manifold topology divergence: a framework for comparing data manifolds. *Advances in neural information processing systems*, 34:7294–7305, 2021b.

Bresten, C. and Jung, J.-H. Detection of gravitational waves using topological data analysis and convolutional neural network: An improved approach. *arXiv preprint arXiv:1910.08245*, 2019.

Bubenik, P. et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

Caliński, T. and Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101.

Chazal, F., Glisse, M., Labruère, C., and Michel, B. Convergence rates for persistence diagram estimation in topological data analysis. In *International Conference on Machine Learning*, pp. 163–171. PMLR, 2014.

Chen, D., Mei, J.-P., Zhang, Y., Wang, C., Wang, Z., Feng, Y., and Chen, C. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7028–7036, 2021a.

Chen, L., Wang, D., Gan, Z., Liu, J., Henao, R., and Carin, L. Wasserstein contrastive representation distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16296–16305, 2021b.

Chen, P., Liu, S., Zhao, H., and Jia, J. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5008–5017, 2021c.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Davies, D. L. and Bouldin, D. W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

Davies, T., Wan, Z., and Sanchez-Garcia, R. J. The persistent laplacian for data science: evaluating higher-order persistent spectral representations of data. In *International Conference on Machine Learning*, pp. 7249–7263. PMLR, 2023.

de Surrel, T., Hensel, F., Carrière, M., Lacombe, T., Ike, Y., Kurihara, H., Glisse, M., and Chazal, F. Ripsnet: a general architecture for fast and robust estimation of the persistent homology of point clouds. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 96–106. PMLR, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Deng, X., Sun, D., Newsam, S., and Wang, P. Distpro: Searching a fast knowledge distillation process via meta optimization. In *European Conference on Computer Vision*, pp. 218–235. Springer, 2022.

Dey, T. K. and Wang, Y. *Computational Topology for Data Analysis*. Cambridge University Press, 2022. doi: 10.1017/9781009099950.

Du, S., Lao, Q., Kang, Q., Li, Y., Jiang, Z., Zhao, Y., and Li, K. Distilling knowledge from topological representations for pathological complete response prediction. In Wang, L., Dou, Q., Fletcher, P. T., Speidel, S., and Li, S. (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pp. 56–65, Cham, 2022. Springer Nature Switzerland.

Girshick, R. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Guo, Z., Yan, H., Li, H., and Lin, X. Class attention transfer based knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11868–11877, 2023.

Güzel, İ. and Kaygun, A. A new non-archimedean metric on persistent homology. *Computational Statistics*, 37 (4):1963–1983, 2022. ISSN 1613-9658. doi: 10.1007/s00180-021-01187-z.

Hajij, M., Zamzmi, G., and Batayneh, F. TDA-Net: Fusion of persistent homology and deep learning features for COVID-19 detection from chest X-Ray images. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 4115–4119, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., and Choi, J. Y. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1921–1930, 2019.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hu, C., Li, X., Liu, D., Wu, H., Chen, X., Wang, J., and Liu, X. Teacher-student architecture for knowledge distillation: A survey. *arXiv preprint arXiv:2308.04268*, 2023.

Hu, X., Samaras, D., and Chen, C. Learning probabilistic topological representations using discrete morse theory. In *The Eleventh International Conference on Learning Representations*, 2022.

Huang, T., You, S., Wang, F., Qian, C., and Xu, C. Knowledge distillation from a stronger teacher. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 33716–33727. Curran Associates, Inc., 2022.

Jeon, E. S., Choi, H., Shukla, A., Wang, Y., Lee, H., Buman, M. P., and Turaga, P. Topological persistence guided knowledge distillation for wearable sensor data. *Engineering Applications of Artificial Intelligence*, 130:107719, 2024. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.107719.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-100 (canadian institute for advanced research).

Le, Y. and Yang, X. S. Tiny imagenet visual recognition challenge. 2015. URL https://api.semanticscholar.org/CorpusID:16664790.

Li, X., Wu, J., Fang, H., Liao, Y., Wang, F., and Qian, C. Local correlation consistency for knowledge distillation. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (eds.), *Computer Vision – ECCV 2020*, pp. 18–33, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58610-2.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection, 2017.

Liu, X., Li, L., Li, C., and Yao, A. Norm: Knowledge distillation via n-to-one representation matching. *arXiv preprint arXiv:2305.13803*, 2023.

Liu, Y., Cao, J., Li, B., Yuan, C., Hu, W., Li, Y., and Duan, Y. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Maria, C., Boissonnat, J.-D., Glisse, M., and Yvinec, M. The gudhi library: Simplicial complexes and persistent homology. In Hong, H. and Yap, C. (eds.), *Mathematical Software – ICMS 2014*, pp. 167–174, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. ISBN 978-3-662-44199-2.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Nawar, A., Rahman, F., Krishnamurthi, N., Som, A., and Turaga, P. Topological descriptors for parkinson's disease classification and regression analysis. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 793–797. IEEE, 2020.

Pachauri, D., Hinrichs, C., Chung, M. K., Johnson, S. C., and Singh, V. Topology-based kernels with application to inference problems in alzheimer's disease. *IEEE transactions on medical imaging*, 30(10):1760–1770, 2011.

Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Passalis, N. and Tefas, A. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284, 2018.

Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., and Zhang, Z. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Peng, Y., Wang, H., Sonka, M., and Chen, D. Z. Phg-net: Persistent homology guided medical image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7583–7592, January 2024.

Purvine, E., Brown, D., Jefferson, B., Joslyn, C., Praggastis, B., Rathore, A., Shapiro, M., Wang, B., and Zhou, Y. Experimental observations of the topology of convolutional neural network activations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9470–9479, 2023.

Reininghaus, J., Huber, S., Bauer, U., and Kwitt, R. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4741–4748, 2015.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2015.

Rousseeuw, P. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, nov 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL https://doi.org/10.1016/0377-0427(87)90125-7.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pp. 4510–4520, 2018.

Som, A., Thopalli, K., Ramamurthy, K. N., Venkataraman, V., Shukla, A., and Turaga, P. Perturbation robust representations of topological persistence diagrams. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 617–635, 2018.

Somasundaram, E. V., Brown, S. E., Litzler, A., Scott, J. G., and Wadhwa, R. R. Benchmarking r packages for calculation of persistent homology. *The R journal*, 13(1):184, 2021.

Stucki, N., Paetzold, J. C., Shit, S., Menze, B., and Bauer, U. Topologically faithful image segmentation via induced matching of persistence barcodes. In *International Conference on Machine Learning*, pp. 32698–32727. PMLR, 2023.

Tian, Y., Krishnan, D., and Isola, P. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Trofimov, I., Cherniavskii, D., Tulchinskii, E., Balabin, N., Burnaev, E., and Barannikov, S. Learning topology-preserving data representations. In *The Eleventh International Conference on Learning Representations*, 2022.

Tung, F. and Mori, G. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

Umeda, Y. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.

Von Rohrscheidt, J. and Rieck, B. Topological singularity detection at multiple scales. In *International Conference on Machine Learning*, pp. 35175–35197. PMLR, 2023.

Wang, T., Yuan, L., Zhang, X., and Feng, J. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4933–4942, 2019.

Wang, Y., Cheng, L., Duan, M., Wang, Y., Feng, Z., and Kong, S. Improving knowledge distillation via regularizing feature norm and direction. *arXiv preprint arXiv:2305.17007*, 2023.

Yang, C., Zhou, H., An, Z., Jiang, X., Xu, Y., and Zhang, Q. Cross-image relational knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12319–12328, 2022a.

Yang, J., Martinez, B., Bulat, A., and Tzimiropoulos, G. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2020.

Yang, J., Martinez, B., Bulat, A., Tzimiropoulos, G., et al. Knowledge distillation via softmax regression representation learning. International Conference on Learning Representations (ICLR), 2021.

Yang, Z., Li, Z., Shao, M., Shi, D., Yuan, Z., and Yuan, C. Masked generative distillation. In *European Conference on Computer Vision*, pp. 53–69. Springer, 2022b.

Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

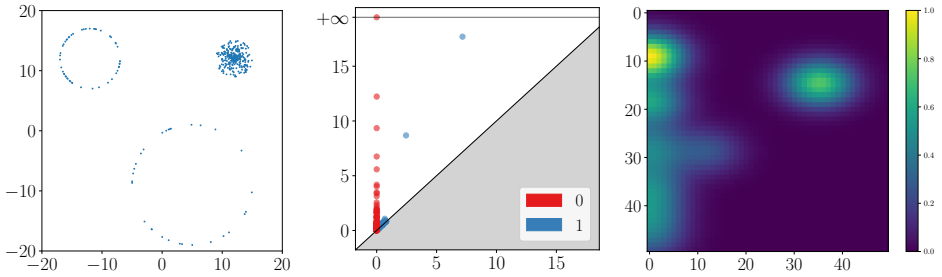
Zagoruyko, S. and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

Zhao, B., Cui, Q., Song, R., Qiu, Y., and Liang, J. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

Zhu, J., Tang, S., Chen, D., Yu, S., Liu, Y., Rong, M., Yang, A., and Wang, X. Complementary relation contrastive distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9260–9269, 2021.

Zong, M., Qiu, Z., Ma, X., Yang, K., Liu, C., Hou, J., Yi, S., and Ouyang, W. Better teacher better student: Dynamic prior knowledge for knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2022.

A. Examples of Persistent Homology



(a) Point cloud data (b) Persistence diagram (c) Persistence image

Figure 6: Examples of persistent homology (PH). (a) Point cloud data (PCD) consisting of two circles and one disk, forming three connected components and two one-dimensional holes. (b) Persistent diagram of the PCD, with three points in 0 dimensions and two points in 1 dimension away from a diagonal line. In (b), the x-axis represents the"birth" times, indicating when specific topological features were born, while the y-axis corresponds to the "death" times, indicating when these features cease to exist. Points far from a diagonal line reflect the global structure of PCD. Points close to the diagonal line reflect the local structure of the PCD and are regarded as noise in this case. (c) A persistence image of the one-dimensional barcode of (b) with a resolution of (50, 50) and a weight function defined by $(death - birth)^2$ to emphasize persistence.

B. Training Details

CIFAR-100. We trained the teacher and student networks from scratch using the He initializer (He et al., 2015). The student networks were trained with the stochastic gradient descent optimizer with a minibatch size of 64 over 240 epochs, and the weight decay was set to $5\mathrm{e}{-4}$ with a momentum of 0.9. For MobileNet (Sandler et al., 2018) and ShuffleNet (Ma et al., 2018), the learning rate was set to 0.01, and for the remaining models, it was set to 0.05, with a decay by a factor of 10 at 150, 180, and 210 epochs. The temperature was set to 4, determined as the optimal value through experiments. We set α to 1 and performed a grid search on β (ranging from 1 to 10) and γ (ranging from 1 to 50) in Eq. (5). All experimental results are reported as the average of five repetitions.

ImageNet-1K. We use the pretrained model provided by PyTorch[1] as the teacher network. The student networks were trained with a minibatch size of 256 over 120 epochs, with the weight decay set to $1\mathrm{e}{-4}$. The initial learning rate was set to 0.1, decreasing by a factor of 10 every 30 epochs. The results on ImageNet-1K are based on a single experimental run. The training details that are not mentioned are consistent with those used for CIFAR-100.

RipsNet. For training RipsNet, 200K PCDs were generated from CIFAR-100 and 20K from ImageNet-1K, with 0.25% used for validation. Additionally, RipsNet was trained over 25K epochs using the Adamax optimizer (Kingma & Ba, 2014), a minibatch size of 64, and a learning rate of $5\mathrm{e}{-4}$. An early stopping was used with a patience of 50. The architecture of RipsNet consists of seven layers, with the rectified linear units activation function for the initial six layers and the sigmoid for the final layer. An operator of RipsNet was chosen with better performance for the mean and sum. The layers within RipsNet can be separated into two distinct segments: layers preceding or subsequent to the operator. The size of the layers in the second segment is the same across all datasets and model pairs, whereas in the first segment, it is determined by the dimensions of the latent space of the teacher network.

C. Extension

To demonstrate the scalability of TopKD, we perform two additional tasks: object detection and transfer learning. In object detection, we evaluate the widely used MS COCO (Lin et al., 2014) dataset, and for transfer learning, we conduct experiments on STL-10 (Coates et al., 2011) and Tiny-ImageNet (Le & Yang, 2015).

[1]https://pytorch.org/vision/main/models.html

C.1. Object Detection

When applying TopKD to the object detection task, we incorporate $\mathcal{L}_{Top}$ for the detector backbone output and $\mathcal{L}_{KD}$ for the box classifier output into the existing loss. The coefficients of each loss are set to 1 and 1.2, respectively. Table 9 presents the results comparing the performance of TopKD on the MS COCO dataset, with the KD, FitNet, FGFI (Wang et al., 2019), DKD, and Review KD models set as the baselines. Consistent with earlier findings, TopKD improves the average precision (AP) performance of the student model by 1.07 on average. In addition, compared to vanilla KD, there are average increases of 0.93, 1.56, and 0.91 for AP, AP_{50}, and AP_{75}, respectively. The experiments are conducted with a batch size of 8 for a fair comparison. This result indicates that the global topology is mimicked using markedly fewer data compared to image classification. Nevertheless, this approach significantly enhances the performance of vanilla KD. The mentioned results suggest that a more pronounced improvement in performance can be attained by designing the object detection task to capture the topology characteristics of the embedding features better.

Table 9: Object detection results based on the Faster R-CNN (Girshick, 2015) with FPN (Lin et al., 2017) on the MS COCO dataset, with AP evaluated in val2017. Tteacher and student pairs are set as ResNet101 (R101) with ResNet18 (R18), ResNet101 with ResNet50 (R50), and ResNet50 with MobileNetV2 (MV2).

	R101 & R18			R101 & R50			R50 & MV2		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Teacher	42.04	62.48	45.88	42.05	62.48	45.88	40.55	61.02	43.81
Student	33.26	53.61	35.26	37.93	58.84	41.05	29.47	48.87	30.90
KD	33.97	54.66	36.62	38.35	59.41	41.71	30.13	50.28	31.35
FitNet	34.13	54.16	36.71	38.76	59.62	41.80	30.00	49.80	31.69
FGFI	35.44	55.51	38.17	39.44	60.27	43.04	31.16	50.68	32.92
DKD	35.05	56.60	37.54	39.25	60.90	42.73	32.34	53.77	34.01
ReviewKD	36.75	56.72	34.00	40.36	60.97	44.08	33.71	53.15	36.13
Ours	34.59	55.54	37.15	39.01	60.54	42.27	31.64	52.69	32.98

C.2. Transfer Learning

We perform transfer learning to assess the generalizability of the feature representation trained with TopKD. In this experiment, we set MobileNetV2 as the student network and ResNet50 as the teacher network. The featurizer of the student model, which was trained on CIFAR-100, is frozen, and only the classifier is finetuned to adapt to the target dataset. The training details are set the same as CIFAR-100. As presented in Table 10, TopKD significantly outperforms vanilla KD, demonstrating its effectiveness in terms of transferability. In addition, TopKD achieves comparable results on both datasets.

Table 10: Comparison of transfer learning from CIFAR-100 to STL-10 and Tiny-ImageNet (Tiny). RevKD denotes ReviewKD.

	Baseline	KD	AT	CRD	RevKD	DKD	CAT-KD	Ours
CIFAR-100 $\longrightarrow$ STL-10	64.39	67.81	65.10	71.46	66.16	71.05	73.20	72.93
CIFAR-100 $\longrightarrow$ Tiny	30.85	32.37	29.13	38.75	32.65	36.48	39.87	35.36

D. Additional Ablation Studies

D.1. Hyperparameters

We conduct grid search to find the optimal coefficients (α, β, γ in Eq. 5) for each component of the total loss. At this time, we fix α to 1 and only adjust the values of β and γ. Tables 11 and 12 present a performance comparison based on these hyperparameters, using ResNet56 as the teacher and ResNet20 as the student, consistent with the ablation studies in Section 5.3. In this setting, the optimal values for β and γ are 2 and 5, respectively. We confirm that performance improved over vanilla KD unless the β value is excessively large. Therefore, we can conclude that TopKD is not sensitive to the weights of each component of the loss function.

Table 11: Performance according to γ (coefficient of L_{Top}).

Teacher		72.34						
Student		69.06						
Vanilla KD		70.66						
TopKD	β	γ						
		1	2	5	7	10	25	50
	2	71.39	71.36	71.58	71.17	71.27	71.32	71.25

Table 12: Performance according to β (coefficient of L_{KD})

Teacher		72.34					
Student		69.06					
Vanilla KD		70.66					
TopKD	γ	β					
		1	2	3	5	7	10
	5	71.16	71.58	71.25	71.15	70.79	70.34

E. Algorithm of TopKD

We present the training algorithms for RipsNet and the student network, as shown in Algorithms 1 and 2.

Algorithm 1 RipsNet training algorithm

Input: The set of PCDs, $P^T = \{p_1, p_2, \cdots, p_n\}$;
Gudhi library for the computation of the exact persistence image, pi;
RipsNet model $\hat{pi}$ with He initialized weights $\hat{\theta}_r$;
Max iterations $Iter$, batch size m
Output: RipsNet model $\hat{pi}$ trained with the target dataset
for $it = 1 \cdots Iter$ **do**
  Sample a random batch $(p_1, p_2, \cdots, p_m)$ and the exact persistence images $pi(p_1, .., p_m)$ from training data P^T;
  Extract the approximated persistence images of $(p_1, p_2, \cdots, p_m)$ from RipsNet, $\hat{pi}(p_1, p_2, \cdots, p_m)$;
  Compute the exact persistence images of $(p_1, p_2, \cdots, p_m)$ from Gudhi, $pi(p_1, p_2, \cdots, p_m)$;
  Train the RipsNet model using $\mathcal{L}_2$ loss:

$$\nabla_{\theta_s}\Big(\sum_{(p_1,..,p_m)\in(P^T)^m} \mathcal{L}_2(\hat{pi}(p_1, .., p_m), pi(p_1, .., p_m))\Big);$$

end for

Algorithm 2 Student model training algorithm

Input: Training dataset, $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$;
Labels of training dataset, $y = y_1, y_2, \cdots, y_n$;
Teacher model F^T with pretrained weights θ_t;
Student model $\hat{F}^S$ with He-initialized weights $\hat{\theta}_s$;
RipsNet model $\hat{pi}$ with pretrained weights θ_r;
Max epoch $Epochs$, batch size m, weight of classification loss α, weight of KD loss β, weight of topology distillation loss γ
Output: Student model F^s trained with optimized weights θ_s;
for $e = 1 \cdots Epochs$ **do**
  Sample a random batch X^m and the corresponding labels Y^m from training data $\mathcal{X}$ and y;
  Extract the embedding features from the teacher featurizer, $(t_1, t_2, \cdots, t_m) = f^T(X^m; \theta_t)$;
  Extract the embedding features and logits from the student featurizer, $(s_1, s_2, \cdots, s_m) = f^S(X^m; \theta_s)$;
  Extract the persistence images of $(t_1, t_2, \cdots, t_m)$ from the pretrained RipsNet, $\hat{pi}(t_1, t_2, \cdots, t_m) = \hat{pi}$;
  Extract the persistence images of $(s_1, s_2, \cdots, s_m)$ from the pretrained RipsNet, $\hat{pi}(s_1, s_2, \cdots, s_m) = \hat{pi}$;
  Transfer the persistence images of the teacher model to the student model by using the stochastic gradient descent from $\mathcal{L}_{Top}$:

$$\nabla_{\theta_s}(\gamma \sum_{(x_1,..,x_m)\in\mathcal{X}^m} \mathcal{L}_2(\hat{pi}(t_1, .., t_m), \hat{pi}(s_1, .., s_m)));$$

  Train the student model using the classification loss and vanilla KD loss:

$$\nabla_{\theta_s}(\alpha\mathcal{L}_{CE} + \beta\mathcal{L}_{KD});$$

end for

F. Computational Complexity and Training Results of RipsNet

F.1. Computational Complexity of RipsNet

To incorporate PH into KD, TopKD includes three additional steps: generating a training dataset composed of PIs for training RipsNet, training RipsNet, and calculating PIs through the pretrained RipsNet during the student model's training. First, the execution time required to produce on PI (Table 13 (1)) depends on the batch size, as the complexity of Vietoris-Rips complex is $O(2^n)$ (Somasundaram et al., 2021), where n represents the number of points. Moreover, since we use only 0-dimensional homology, the initial computation of PD is not computationally demanding. Subsequently, the training time for RipsNet (Table 13 (2)) is merely several minutes. The reduced training time for ImageNet-1K is due to early stopping. Once RipsNet is trained, it serves solely for calculating PIs during the student model's training. Additionally, we calculate the floating point operations (FLOPs) of RipsNet (Table 13 (3)). By approximating PIs, we enable the incorporation of PD into the training loop, thus circumventing the prohibitively large computational complexity required for exact PIs. This ensures that distillation with the topology distillation loss operates successfully without imposing heavy computational overhead.

Table 13: (1) An execution time to produce one PI on AMD Epyc 7742. (2) The training time of RipsNet on an A100 GPU. (3) FLOPs of RipsNet.

Dataset (Teacher, Batch size)	CIFAR-100 (ResNet56, 64)	ImageNet-1K (ResNet34, 256)
(1) Gudhi	1.07ms	22.83ms
(2) Training RipsNet	65.75min	24.80min
(3) FLOPs of RipsNet	1.131M	235.1M

F.2. Training Results of RipsNet

This section provides a description of the hyperparameters used for RipsNet and presents the corresponding training results. Tables 14, 16, 17, and 19 present the results of the CIFAR-100 dataset. Table 15 is based on ImageNet-1K, and Table 18 utilizes the MS COCO dataset. The "unit list" refers to the hidden layer dimensions of RipsNet, and "PD dim" denotes the dimension of the PD used for training RipsNet. The operator (mean or sum) is selected based on achieving lower loss. The "location" indicates the latent space targeted for transfer. As illustrated in Fig. 2(b), the student is trained so that its outputs from the fourth stage match those of the teacher. The corresponding training results are shown in Tables 1, 2, 3 and 4.

Table 14: Training results of RipsNet used in Tables 1 and 2.

Teacher model	Location	Batch Size	PD dim	Operator	Unit list	Best loss
ResNet56	Stage 4	64	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.001176
ResNet56	Stage 4	64	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.001179
ResNet110	Stage 4	64	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.001221
ResNet110	Stage 4	64	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.001229
ResNet32x4	Stage 4	64	0	Mean	[256, 256, 128, 128, 50, 100, 200, 400]	0.001127
ResNet32x4	Stage 4	64	0	Sum	[256, 256, 128, 128, 50, 100, 200, 400]	0.001128
WRN-40-2	Stage 4	64	0	Mean	[128, 128, 64, 64, 50, 100, 200, 400]	0.001155
WRN-40-2	Stage 4	64	0	Sum	[128, 128, 64, 64, 50, 100, 200, 400]	0.001149
VGG13	Stage 4	64	0	Mean	[512, 512, 256, 256, 50, 100, 200, 400]	0.001325
VGG13	Stage 4	64	0	Sum	[512, 512, 256, 256, 50, 100, 200, 400]	0.001313
ResNet50	Stage 4	64	0	Mean	[2048, 1024, 512, 256, 50, 100, 200, 400]	0.001288
ResNet50	Stage 4	64	0	Sum	[2048, 1024, 512, 256, 50, 100, 200, 400]	0.001262

Table 15: Training results of RipsNet used in Tables 3 and 4.

Teacher model	Location	Batch size	PD dim	Operator	Unit list	Best loss
ResNet50	Stage 4	256	0	Mean	[2048, 1024, 512, 256, 50, 100, 200, 400]	0.000814
ResNet50	Stage 4	256	0	Sum	[2048, 1024, 512, 256, 50, 100, 200, 400]	0.000810
ResNet50	Stage 4	256	0	Mean	[512, 512, 256, 256, 50, 100, 200, 400]	0.000655
ResNet50	Stage 4	256	0	Sum	[512, 512, 256, 256, 50, 100, 200, 400]	0.000660

Table 16: Training results of RipsNet used in Table 5.

Teacher model	Location	Batch size	PD dim	Operator	Unit list	Best loss
ResNet56	Stage 4	64	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.001176
ResNet56	Stage 4	64	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.001179
ResNet56	Stage 4	64	1	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.000323
ResNet56	Stage 4	64	1	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.000323
ResNet56	Stage 4	64	0+1	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.000073
ResNet56	Stage 4	64	0+1	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.000072

Table 17: Training results of RipsNet used in Fig. 3.

Teacher model	Location	Batch size	PD dim	Operator	Unit list	Best loss
ResNet56	Stage 4	8	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.002935
ResNet56	Stage 4	8	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.002960
ResNet56	Stage 4	16	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.002372
ResNet56	Stage 4	16	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.002391
ResNet56	Stage 4	32	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.001455
ResNet56	Stage 4	32	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.001467
ResNet56	Stage 4	64	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.001176
ResNet56	Stage 4	64	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.001179
ResNet56	Stage 4	128	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.000831
ResNet56	Stage 4	128	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.000833
ResNet56	Stage 4	256	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.000570
ResNet56	Stage 4	256	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.000568

Table 18: Training results of RipsNet used in Table 9.

Teacher-Student models	Location	Batch size	PD dim	Operator	Unit list	Best loss
R101-R18	Backbone	8	0	Mean	[256, 256, 128, 128, 50, 100, 200, 400]	0.000418
R101-R18	Backbone	8	0	Sum	[256, 256, 128, 128, 50, 100, 200, 400]	0.000431
R101-R50	Backbone	8	0	Mean	[256, 256, 128, 128, 50, 100, 200, 400]	0.000569
R101-R50	Backbone	8	0	Sum	[256, 256, 128, 128, 50, 100, 200, 400]	0.000595
R50-MV2	Backbone	8	0	Mean	[256, 256, 128, 128, 50, 100, 200, 400]	0.000501
R50-MV2	Backbone	8	0	Sum	[256, 256, 128, 128, 50, 100, 200, 400]	0.000535

Table 19: Training results of RipsNet used in Table 7.

Teacher model	Location	Batch size	PD dim	Operator	Unit list	Best loss
ResNet56	Stage 1	64	0	Mean	[16, 16, 16, 16, 50, 100, 200, 400]	0.000916
ResNet56	Stage 1	64	0	Sum	[16, 16, 16, 16, 50, 100, 200, 400]	0.000927
ResNet56	Stage 2	64	0	Mean	[16, 16, 16, 16, 50, 100, 200, 400]	0.000842
ResNet56	Stage 2	64	0	Sum	[16, 16, 16, 16, 50, 100, 200, 400]	0.000847
ResNet56	Stage 3	64	0	Mean	[32, 32, 16, 16, 50, 100, 200, 400]	0.000876
ResNet56	Stage 3	64	0	Sum	[32, 32, 16, 16, 50, 100, 200, 400]	0.000885
ResNet56	Stage 4	64	0	Mean	[64, 64, 32, 32, 50, 100, 200, 400]	0.001176
ResNet56	Stage 4	64	0	Sum	[64, 64, 32, 32, 50, 100, 200, 400]	0.001179

G. Additional Results of Numerical Analysis

G.1. Experimental Setting for Measuring Approximation Errors of RipsNet

We evaluate the approximation errors of RipsNet for both the teacher and student networks. For each minibatch of 64 training samples, we measure the $\mathcal{L}_2$ distance between the exact and approximated PIs of embedding features. Here, we note that even though RipsNet is trained using embedding features of training samples from the teacher network, it never observes the embedding features of the student network. Hence, it is crucial for RipsNet to accurately approximate PIs for the training samples throughout the training process of the student network. For this reason, we present the approximation errors for training samples in Table 8.

G.2. Similarity Between PDs of Teacher and Student Networks

Even though we define PD of embedding features as global topology knowledge over a minibatch, our method aims to reduce the $\mathcal{L}_2$ distance between the approximated PIs of the teacher and student networks for practical implementation. However, there is no theoretical guarantee that the converged student network produces embedding features that have similar PDs to those of the teacher network. Therefore, we measure the ∞-Wasserstein distance between the 0-dim PDs of embedding features from the teacher and student networks. We use the input activations of the classifiers for each minibatch of 32 test samples from the CIFAR-100 dataset. As indicated in Table 20, our method does not consistently reduce the Wasserstein distance between the PDs of the teacher and student networks. Nonetheless, the results suggest that further performance improvement could be achieved by investigating a way of directly matching the PDs of embedding features from the teacher and student networks.

Table 20: Wasserstein distance between the PDs of embedding features from the teacher and student networks. For each minibatch of 32 test samples from CIFAR-100, we compute the ∞-Wasserstein distance between 0-dim PDs of the embedding features from the teacher and student networks. The values are averaged across minibatches. For each setting, the value closest to the teacher network is bolded.

Teacher	Student	Student	Student w/ KD	Student w/ TopKD
ResNet56	ResNet20	1.1737	2.3912	**0.6971**
ResNet110	ResNet20	3.6203	**1.3735**	2.4723
ResNet110	ResNet32	0.5689	**0.2757**	0.4606
ResNet32x4	ResNet8x4	1.2387	**1.0938**	1.5949
WRN-40-2	WRN-16-2	3.1489	1.1091	**1.0184**
WRN-40-2	WRN-40-1	1.5055	1.7722	**1.3374**
VGG13	VGG8	1.0598	1.8062	**0.9832**
VGG13	MobileNetV2	5.4027	**0.9630**	3.5622
ResNet50	MobileNetV2	**2.2088**	3.0111	2.3972
ResNet32x4	ShuffleNetV1	4.3594	**3.3245**	5.0185
ResNet32x4	ShuffleNetV2	**3.1055**	4.1220	3.4549
WRN-40-2	ShuffleNetV1	**2.2203**	2.7840	3.2101

G.3. Quantitative Analysis for UMAP Visualizations

In this section, we conduct a quantitative analysis of the UMAP visualizations presented in Fig. 5. To assess the separability of the embedding features across various classes, we adopt three widely used metrics for evaluating clustering algorithms: Silhouette (Rousseeuw, 1987), Calinski-Harabasz index (Caliński & Harabasz, 1974), and Davies-Bouldin index (Davies & Bouldin, 1979). These metrics compare within-cluster distances and between-cluster distances. For the evaluation, we randomly select 10 classes from the CIFAR-100 test dataset and compute these metrics for the 2-dimensional vectors generated by the UMAP algorithms. This procedure is repeated 10 times with different seeds, and the results are summarized in Table 21, which presents the mean and standard deviation of the three metrics. Notably, our TopKD demonstrates superior discriminability across different classes in the UMAP visualizations.

Table 21: Quantitative analysis of the UMAP visualizations. VGG13 is used for the teacher network, and MobileNetV2 for the student network. Higher values indicate better performance for the Silhouette score and the Calinski-Harabasz index, while lower values are preferable for the Davies-Bouldin index.

	Silhouette(↑)	Calinski-Harabasz index(↑)	Davies-Bouldin index(↓)
Teacher	0.336 (0.034)	654.618 (94.885)	1.139 (0.188)
Student	0.251 (0.046)	460.893 (70.930)	1.598 (0.599)
Student w/ KD	0.208 (0.050)	374.757 (53.336)	1.732 (0.546)
Student w/ TopKD	0.297 (0.050)	542.665 (95.834)	1.404 (0.427)

G.4. Additional Comparisons of Exact and Approximated PIs with Different Settings

Figs. 7 and 8 show the exact and approximated PIs for the embedding features of minibatches extracted from the teacher and student networks. Each column corresponds to a minibatch consisting of 64 test samples randomly sampled from the CIFAR-100 dataset.

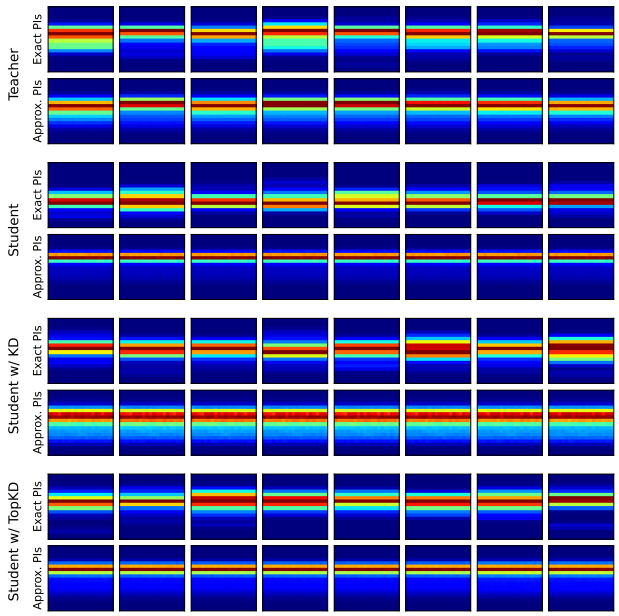


Figure 7: Exact and approximated PIs for the embedding features of minibatches extracted from the teacher network (ResNet50) and the student network (VGG8). Each column corresponds to a minibatch with a size of 64.

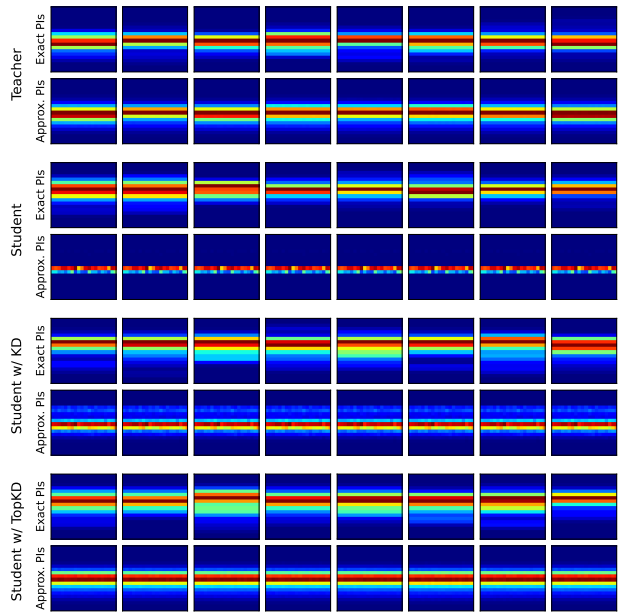


Figure 8: Exact and approximated PIs for the embedding features of minibatches extracted from the teacher network (WRN-40-2) and the student network (ShuffleNetV1). Each column corresponds to a minibatch with a size of 64.

G.5. Additional Visualizations of Overall Topology

To provide further insights through visualization, we perform the additional UMAP visualization experiments on CIFAR-100 and ImageNet-1K datasets. In these experiments, we sampled 10 classes randomly each time. Figs. 9 and 10 display the results for CIFAR-100, while Figs. 11 and 12 present the results for ImageNet-1K. Consistently, TopKD exhibits superior performance in clustering compared to the student network and vanilla KD and the boundaries between each class are clear. In the case of the ImageNet-1K experiments, TopKD shows a greater concentration of embedding features for each class.

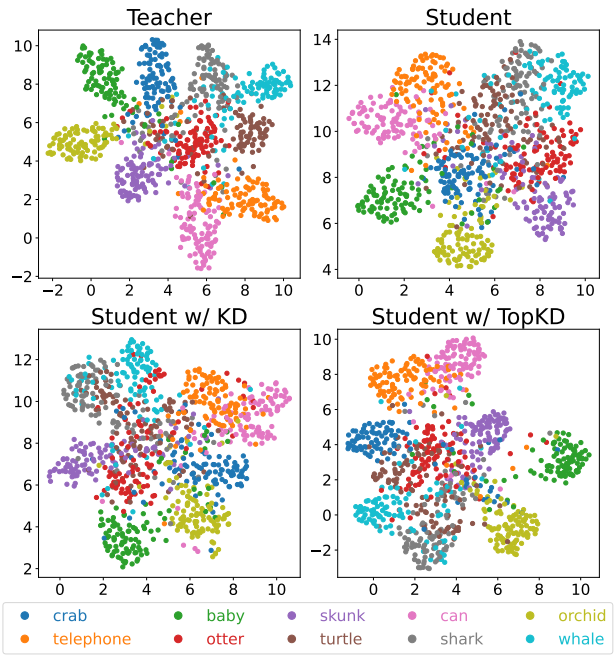


Figure 9: UMAP visualizations of embedding features of featurizers of the teacher (VGG13) and student (MobileNetV2) networks. We randomly select 10 classes from the test data of CIFAR-100.

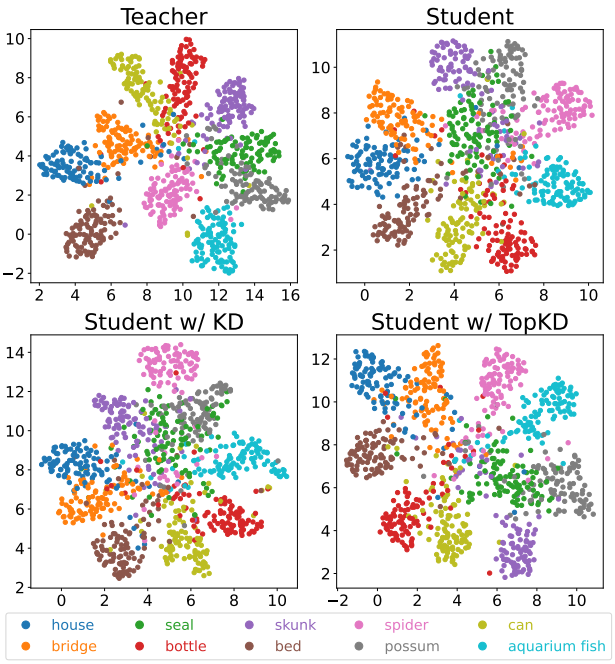


Figure 10: UMAP visualizations of embedding features of featurizers of the teacher (VGG13) and student (MobileNetV2) networks. We randomly select 10 classes from the test data of CIFAR-100.

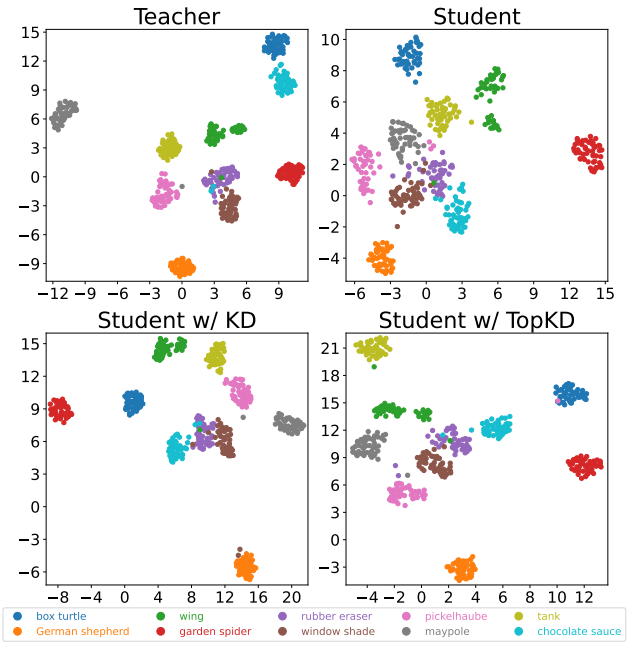


Figure 11: UMAP visualizations of embedding features of featurizers of the teacher (ResNet34) and student (ResNet18) networks. We randomly select 10 classes from the test data of ImageNet-1K.

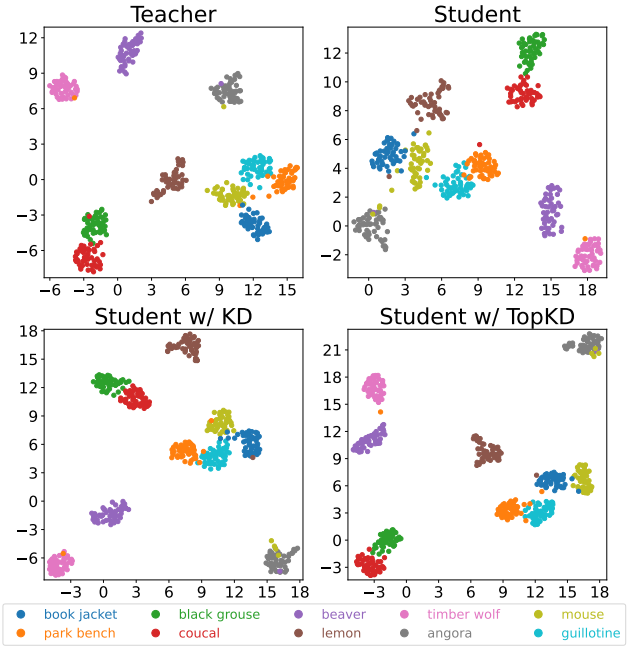


Figure 12: UMAP visualizations of embedding features of featurizers of the teacher (ResNet34) and student (ResNet18) networks. We randomly select 10 classes from the test data of ImageNet-1K.