Knowledge Distillation Using Hierarchical Self-Supervision Augmented Distribution

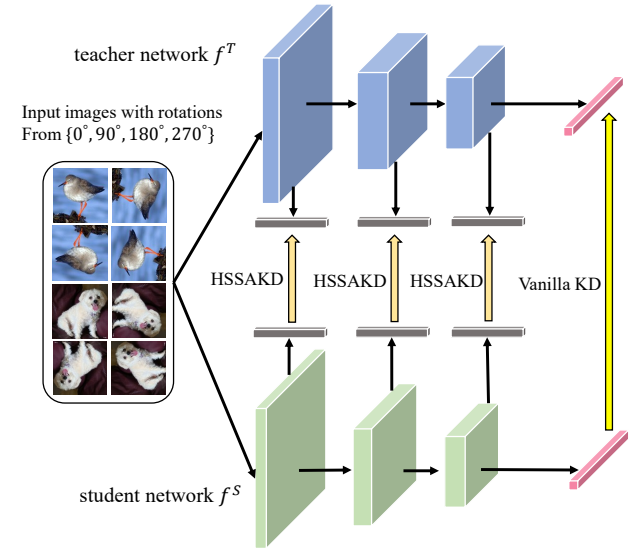
Chuanguang Yang, Zhulin An, Linhang Cai, and Yongjun Xu

***Abstract*—Knowledge distillation (KD) is an effective framework that aims to transfer meaningful information from a large teacher to a smaller student. Generally, KD often involves how to define and transfer knowledge. Previous KD methods often focus on mining various forms of knowledge, for example, feature maps and refined information. However, the knowledge is derived from the primary supervised task and thus is highly task-specific. Motivated by the recent success of self-supervised representation learning, we propose an auxiliary self-supervision augmented task to guide networks to learn more meaningful features. Therefore, we can derive soft self-supervision augmented distributions as richer dark knowledge from this task for KD. Unlike previous knowledge, this distribution encodes joint knowledge from supervised and self-supervised feature learning. Beyond knowledge exploration, we propose to append several auxiliary branches at various hidden layers, to fully take advantage of hierarchical feature maps. Each auxiliary branch is guided to learn self-supervision augmented task and distill this distribution from teacher to student. Overall, we call our KD method as Hierarchical Self-Supervision Augmented Knowledge Distillation (HSSAKD). Experiments on standard image classification show that both offline and online HSSAKD achieves state-of-the-art performance in the field of KD. Further transfer experiments on object detection further verify that HSSAKD can guide the network to learn better features. The code is available at https://github.com/winycg/HSAKD.**

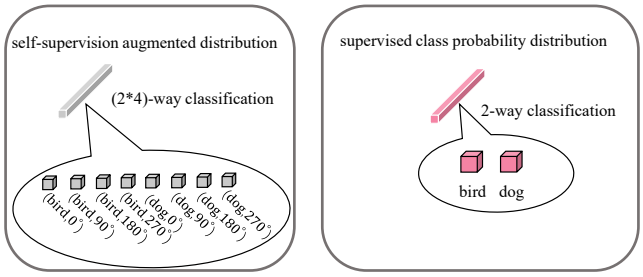
***Index Terms*—Knowledge Distillation, Self-Supervised Learning, Representation Learning, Visual Recognition**

I. Introduction

ALTHOUGH deep CNNs have achieved great successes over a series of visual tasks [1], [2], [3], [4], it is often hard to deploy over-parameterized networks on resource-limited edge devices. Typical solutions can be efficient architecture design [5], [6], [7], [4], [8], model pruning [9], [10], [11] and knowledge distillation [12], [13], [14], [15], [16], [17]. Knowledge Distillation (KD) provides a simple framework that transfers knowledge from a pre-trained high-capacity teacher network to a smaller and faster student network. Compared with independent training, the student network can generalize better benefiting from the additional guidance. The current pattern of KD can be summarized as two critical aspects: (1) what kind of knowledge encapsulated in teacher network can be explored for KD; (2) How to effectively transfer knowledge from teacher to student.



(a) Hierarchical Self-Supervision Augmented KD



(b) Legend of distilled knowledge types.

Fig. 1: An overview of our proposed HSSAKD and distilled knowledge types.

The original KD [12] minimizes the KL-divergence of predictive class probability distributions between student and teacher networks, which makes intuitive sense to force the student to mimic how a superior teacher generates the final predictions. However, such highly abstract dark knowledge ignores much comprehensive information encoded in hidden layers. Later works naturally proposed to transfer feature maps [18] or their refined information [19], [20], [21] between intermediate layers of teacher and student. A reasonable interpretation of the success of feature-based distillation lies in that hierarchical feature maps throughout the CNN represent the intermediate learning process with inductive biases of the final solution. Beyond knowledge alignment limited in the in-

Chuanguang Yang and Linhang Cai are with Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China and University of Chinese Academy of Sciences, Beijing 100049, China. Email: {yangchuanguang, cailinhang19g}@ict.ac.cn

Zhulin An and Yongjun Xu are with Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. Email: {anzhulin, xyj}@ict.ac.cn

Zhulin An is the corresponding author.

Manuscript received July 31, 2021; revised May 19, 2022; accepted June 20, 2022.

dividual sample, more recent works [22], [23] leverage cross-sample correlations or dependencies in high-layer feature embedding space. A common characteristic of previous works often mines knowledge only derived from a single supervised classification task. The task-specific knowledge may hinder the network from exploring richer feature representations.

To excavate more meaningful knowledge, it makes sense to introduce an auxiliary task to guide the network to produce extra representational information, complementary to the primary supervised classification knowledge. From the perspective of representation learning, a natural method is to consider an extra self-supervised task. The idea behind self-supervised learning is to learn *general feature representations* guided by artificial supervisions derived from data itself via a pre-defined pretext task. Generally, learned features could well generalize to other semantic tasks, *e.g.* image classification [24], [25]. Some self-supervised learning works can also be seen as a classification problem with artificial label space, *e.g.* random rotations [24] and jigsaw puzzles with shuffled permutations [25]. By forcing a network to recognize which transformation is applied on an input transformed image, the network would learn general semantic features.

Based on the fact that both supervised and self-supervised tasks are classification problems, we propose to combine two label spaces via a Cartesian product and model full-connected interactions, inspired by [26]. Formally, given a N-way supervised task and a M-way self-supervised task, we merge them into a $N * M$-way joint task, where each label is a 2-tuple that consists of one supervised label and one self-supervised label. We name this joint task as ***self-supervision augmented task*** due to that supervised classification is our primary task. Based on this task, we can naturally derive a joint class probability distribution as a new form of dark knowledge. For ease of notation, we name this knowledge as ***self-supervised augmented distributions***. Fig. 1 illustrates a simple example of the joint label space. Through learning and distilling self-supervision augmented knowledge, the network can benefit from supervised and self-supervised representation learning jointly.

Another valuable problem lies in how to effectively learn and transfer the probabilistic knowledge between teacher and student. Response-based KD [12] aligns probability distributions only in the final layer, because it may be difficult to explicitly derive comprehensive probability distributions from hidden feature maps over the original architecture. Inspired by deeply-supervised nets (DSN) [27], a natural idea is to append several auxiliary branches to a network at various hidden layers. As illustrated in Fig. 1, those branches enable the network to generate multi-level self-supervision augmented distributions from hierarchical feature maps, which further allows us to perform one-to-one distillation in hidden layers towards probabilistic knowledge. However, designing a suitable auxiliary branch to learn an auxiliary task is also a non-trivial problem. Unlike a simple linear classifier in DSN, our auxiliary branch includes an extra feature extraction module. The module is responsible for transforming the feature map derived from the backbone network to adapt for a new task.

Given a single network, we guide the backbone network to learn the primary supervised task and all auxiliary branches to learn our self-supervision augmented task. Surprisingly, this method significantly improves the backbone network performance over the primary supervised task compared to the baseline, *e.g.* an average accuracy gain of 4.15% across widely used network architectures on CIFAR-100. We attribute the gain to our auxiliary self-supervision augmented task that can guide the backbone network to learn better feature representations.

Based on the success of the auxiliary self-supervision augmented task, we think it is promising to distill self-supervision augmented distributions towards all auxiliary branches in a one-to-one manner. Because distributions at various depths are derived from hierarchical feature maps respectively, we name our method as ***Hierarchical Self-Supervision Augmented Knowledge Distillation*** (HSSAKD). For offline HSSAKD, we first train a teacher network using the self-supervision augmented task and then transfer self-supervision augmented distributions as soft labels to supervise a student network. For online HSSAKD, we perform mutual distillation of self-supervision augmented distributions among several networks jointly trained from scratch. Note that all auxiliary branches are only used to assist knowledge transfer and would be dropped at the inference period, leading to no extra costs compared with the original network.

Experimental results on standard CIFAR-100 and ImageNet demonstrate that both offline and online HSSAKD can significantly achieve better performance than other competitive KD methods. Fig. 1 illustrates the superiority over SOTA CRD [23] and SSKD [28] with large accuracy gains. Extensive results on transfer experiments to classification and detection suggest that HSSAKD can teach a network to learn better general features for semantic recognition tasks.

This paper is an extension of our conference version [29]. The new contributions are summarized as follows:

- We examine the effectiveness of several variants involving supervised as well as self-supervised tasks and their derived knowledge for training and KD. We demonstrate that our self-supervision augmented task is a powerful auxiliary task for representation learning.
- Based on the success of offline HSSAKD, we extend it to online HSSAKD for distilling self-supervision augmented distributions mutually. To our knowledge, this method may be the first work that utilizes self-supervised representation knowledge for online KD.
- Our online HSSAKD also achieves SOTA performance on standard image classification benchmarks and significantly surpasses other competitive online KD methods.
- More systematical algorithm descriptions, more detailed ablation study and analyses for offline and online HSSAKD are presented.

II. Related Work

A. Offline Knowledge Distillation

The conventional offline KD conducts a two-stage training pipeline. It first trains a powerful teacher network and transfers knowledge from the pre-trained teacher to a smaller and

faster student network via a meaningful distillation strategy. The seminal KD [12] popularizes the pattern of knowledge transfer with a soft class posterior probability distribution. Later methods further explore feature-based information encapsulated in hidden layers for KD, such as intermediate feature maps [18], attention maps [19], gram matrix [30] and activation boundaries [20]. Unlike exploiting features from hidden layers, more recent works model cross-sample high-order relationships using high-level abstract feature embeddings for KD. Typical metrics of relationships between various samples can be pairwise distance [31], correlation congruence [22] and similarity graph [32]. Based on this idea, Ye *et al.* [33] distill cross-task knowledge via high-order relationship matching. Unlike explicitly transferring relationships, CRD [23] applies contrastive learning among samples from both the teacher and student networks to implicitly perform contrastive representation distillation.

Inspired by recent self-supervised learning [34], SSKD [28] extracts structured knowledge from an auxiliary self-supervised task. Although both SSKD and our HSSAKD take advantage of self-supervised learning, the difference lies in how to leverage self-supervision to define knowledge. SSKD models contrastive relationships by forcing various transformed versions with rotations over the same image closed. As verified in [29], this contrastive distribution may destroy the original visual semantics. In contrast, our HSSAKD explores a more informative knowledge form by encoding supervised and self-supervised distillation into a unified distribution. It does not interfere with a network in recognizing the original semantics, which is very different from SSKD.

Beyond knowledge mining, another aspect is to examine the transfer strategy. The conventional methods [18], [19], [30] often use L2 loss to align feature maps. Moreover, VID [21] maximizes the mutual information between matched features. Wang *et al.* [35] utilize the idea of generative adversarial network to make feature distributions similar. PKT [36] uses different kernels to estimate the probability distribution along with adaptive divergence metrics for KD. Mirzadeh *et al.* [37] point out that a large capability gap between teacher and student may lead to a negative supervisory efficacy. To bridge the gap, TAKD [37] inserts an intermediate-sized network as a teacher assistant and performs multi-step KD. With the same motivation, Passalis *et al.*[38] introduce a proxy model to "explaining" knowledge teacher works to the student. However, extra teacher models increase the complexity of the training pipeline. We append several well-designed auxiliary branches to alleviate the knowledge gap. It facilitate comprehensive knowledge transfer from hierarchical feature maps.

B. Online Knowledge Distillation

Unlike the conventional offline KD, online KD has not a specific teacher-student role and trains multiple student networks simultaneously to learn from each other in a peer-teaching manner. DML [39] indicates that a cohort of student networks can benefit from transferring class probability distributions mutually in an online manner. CL-ILR [40] extends this framework to a hierarchical network with multiple classification heads. In contrast to a mutual mechanism, ONE [41] produces gated ensemble logits as a teacher role for online KD. Chen *et al.* [42] point out the homogenization problem in ONE and utilize a self-attention module to compute aggregation weights for ensemble logits. With the same motivation, KDCL [43] proposes to adaptively aggregate logits via a principle manner. Beyond class probabilities, AFD [44] performs mutual mimicry between feature distributions via an online adversarial training framework. Yang *et al.* [45], [46] apply contrastive learning among multiple peer networks for online KD. Our HSSAKD can also be extended to online KD and perform mutual mimicry of hierarchical self-supervised augmented distributions. To our knowledge, HSSAKD may be the first work that utilizes self-supervised representation knowledge for online KD.

C. Self-Supervised Learning

Self-supervised learning has been widely applied for computer vision, such as image classification [34], [47], [48], video recognition [49], [50] and human pose estimation [51]. The basic idea of self-supervised learning is to design a pretext task that can guide the network to learn meaningful feature representations. Some seminal self-supervised learning works can be regarded as a classification framework that trains a network to learn which transformation is applied to a transformed input image. In this vein, typical transformations can be rotations [52], jigsaw [24] and colorization [53]. Another vein is to use contrastive learning. More recently, SimCLR [34] and PIRL [47] learn invariant feature representations by maximizing the consistency among various transformed versions of the same image. We remark that both SSKD and our HSSAKD refer to self-supervised works to define knowledge. SSKD [28] follows SimCLR [34] to extract self-supervised contrastive-based relationships. In contrast, HSSAKD combines the supervised and self-supervised tasks as a joint classification task and aims to transfer richer joint knowledge.

III. Methodology

A. Self-Supervision Augmented Distribution

In this section, we describe the details of our proposed self-supervision augmented distribution. Before this, we first review the conventional *supervised classification task*, *self-supervised classification task* and *multi-task learning*. The overview is illustrated in Fig. 2.

1) N-way supervised classification task: We consider a training set $\mathcal{X}$ with fully-annotated label space $\mathcal{Y} = \{1, 2, \cdots, N\}$, where $\boldsymbol{x} \in \mathcal{X}$ is an input sample with the label $y \in \mathcal{Y}$, N is the number of classes. We aim to train an ordinary CNN $f(\cdot)$ to map the input sample $\boldsymbol{x}$ to a predictive class probability distribution $\boldsymbol{p} \in \mathbb{R}^N$, *i.e.* $f : \boldsymbol{x} \rightarrow \boldsymbol{p}$. $f(\cdot)$ could be decomposed of a feature extractor $\Phi(\cdot)$ and a linear classifier W. Given an input $\boldsymbol{x}$, $\Phi(\cdot)$ encodes it into a feature embedding vector $\Phi(\boldsymbol{x}) \in \mathbb{R}^d$, where d is the embedding size. Afterwards, a linear matrix $W \in \mathbb{R}^{d \times N}$ transforms the feature vector $\Phi(\boldsymbol{x}) \in \mathbb{R}^d$ into a class logits distribution $f(\boldsymbol{x}) = W^T \Phi(\boldsymbol{x}) \in \mathbb{R}^N$. Following the common protocol, we

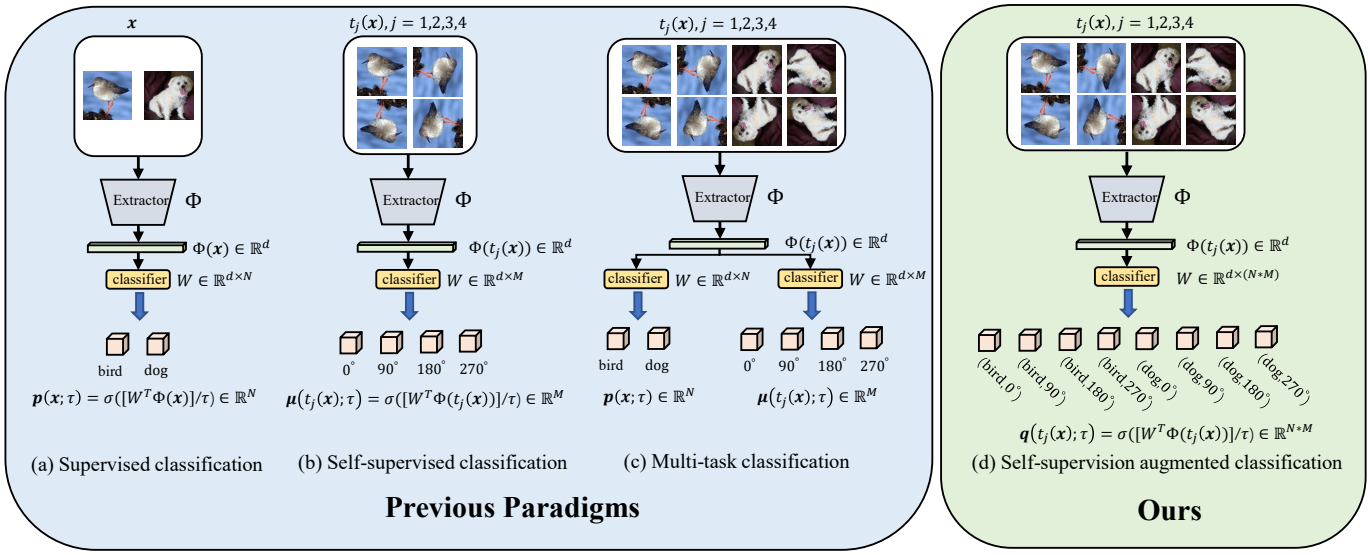


Fig. 2: An overview of (a) N-way supervised classification, (b) M-way self-supervised classification, (c) multi-task learning by performing N-way supervised and M-way self-supervised classification respectively, (d) $(N * M)$-way self-supervision augmented classification. Here, we employ (N=2)-way supervised {bird, dog} and (M=4)-way self-supervised $\{0°, 90°, 180°, 270°\}$ classification for illustration.

scale the logits distribution $f(\boldsymbol{x})$ with a temperature hyper-parameter τ and then normalize it by a *softmax* function σ to derive the final class probability distribution:

$$\boldsymbol{p}(\boldsymbol{x};\tau) = \sigma(f(\boldsymbol{x})/\tau) = \sigma([W^T\Phi(\boldsymbol{x})]/\tau) \in \mathbb{R}^N. \quad (1)$$

2) M-way self-supervised classification task: In principle, the self-supervised task can be regarded as a classification problem to deal with human pre-defined transformations. This framework has been successfully applied to self-supervised feature representation learning, *i.e.* random rotations [25] and colorful image colorization [53]. Formally, we define M various image transformations $\{t_j\}_{j=1}^M$ over an input image $\boldsymbol{x}$ with the label space $\mathcal{M} = \{1, \cdots, M\}$, where t_1 denotes the identity transformation, *i.e.* $t_1(\boldsymbol{x}) = \boldsymbol{x}$. We encourage the network to correctly recognize which transformation t_j is applied to the transformed image. Given an input sample $t_j(\boldsymbol{x})$, the final output is a M-way class probability distribution denoted as:

$$\boldsymbol{\mu}(t_j(\boldsymbol{x};\tau)) = \sigma(f(t_j(\boldsymbol{x})/\tau) = \sigma([W^T\Phi(t_j(\boldsymbol{x}))]/\tau) \in \mathbb{R}^M. \quad (2)$$

Here, $W \in \mathbb{R}^{d\times M}$ is the linear classifier for M-way self-supervised classification.

3) Multi-task learning: To learn feature representations benefiting from both supervised and self-supervised tasks, a natural idea is to employ multi-task learning. In practice, we can use a shared feature extractor and separated classifier heads to perform N-way supervised and M-way self-supervised classification, respectively.

4) $(N * M)$-way self-supervision augmented classification: Although we aim to solve a supervised classification task, we can further introduce an additional self-supervised task to enrich the class space. We combine supervised and self-supervised tasks into a unified task to guide the network learning a joint distribution. The label space of this unified task can be expressed as $\mathcal{N} \otimes \mathcal{M}$, where $\otimes$ is a Cartesian product. $|\mathcal{N} \otimes \mathcal{M}| = N * M$, where $|\cdot|$ is the cardinality of the label collection and * denotes element multiplication. Intuitively, a more competitive joint task would require the network to generate more informative and meaningful feature representations compared with a single task. Given a transformed input image $t_j(\boldsymbol{x}), j \in \mathcal{M}$, Similar to the practice of the conventional classification problem, we can derive the final self-supervision augmented distribution as:

$$\boldsymbol{q}(t_j(\boldsymbol{x});\tau) = \sigma(f(t_j(\boldsymbol{x}))/\tau) = \sigma([W^T\Phi(t_j(\boldsymbol{x}))]/\tau) \in \mathbb{R}^{N*M} \quad (3)$$

over the joint label space $\mathcal{N} \otimes \mathcal{M}$. Here, $W \in \mathbb{R}^{d\times(N*M)}$ is the linear classifier for $N * M$-way self-supervision augmented classification. In this paper, we find our proposed self-supervision augmented distribution is the best type of probabilistic knowledge to learn or conduct KD-based feature representation learning.

B. Auxiliary Architecture Design

It has been widely known that a general CNN gradually downsamples the feature maps along the spatial dimension to refine meaningful semantics. Feature maps with various spatial resolutions produced from different convolutional stages often encode various patterns of representational information. Higher-resolution feature maps from low-level stages often present more fine-grained object details and high-frequency information favourable for small object detection and fine-grained image classification. In contrast, lower-resolution ones from high-level stages extracted by a larger receptive field often contain richer global structures and semantic information. They may lead to better recognition performance by reducing the redundant details. Leveraging multi-scale feature

TABLE I: Architectural details of a ResNet-56 [1] as the backbone $f(\cdot)$ with its three auxiliary branches $\{c_l(\cdot)\}_{l=1}^{3}$. We force the backbone $f(\cdot)$ to output the primary N-way supervised class probability distribution and three auxiliary branches $c_1(\cdot) \sim c_3(\cdot)$ to output $(N*M)$-way self-supervision augmented distributions.

Layer Name	Spatial Size	$f(\cdot)$	$c_1(\cdot)$	$c_2(\cdot)$	$c_3(\cdot)$
Stem	32×32	$3\times3, 16$	-	-	-
Stage1	32×32	$\begin{bmatrix}3\times3,16\\3\times3,16\end{bmatrix}\times 9$	-	-	-
Stage2	16×16	$\begin{bmatrix}3\times3,32\\3\times3,32\end{bmatrix}\times 9$	$\begin{bmatrix}3\times3,32\\3\times3,32\end{bmatrix}\times 9$	-	-
Stage3	8×8	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times 9$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times 9$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times 9$	-
-	8×8	-	-	-	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times 9$
Pooling	1×1	Global Average Pooling			
Linear Classifier	1×1	N-D FC	$(N*M)$-D FC	$(N*M)$-D FC	$(N*M)$-D FC

information is demonstrated as an effective practice in previous works [54], [55]. To take full advantage of hierarchical feature maps encapsulated in a single network, we append several intermediate auxiliary branches into hidden layers. These branches are responsible for generating and distilling multi-level self-supervision augmented distributions from multi-scale feature maps.

Modern CNNs typically utilize stage-wise convolutional blocks to gradually extract coarser features as the depth of the network increases. For example, the popular ResNet-50 for ImageNet classification contains consecutive four stages, and extracted feature maps produced from various stages have different granularities and patterns. Formally, we describe how to insert auxiliary branches into a conventional CNN in a heuristic principle. Given a CNN $f(\cdot)$ that consists of L stages, we choose to append an auxiliary branch after each stage, thus resulting in L branches $\{c_l(\cdot)\}_{l=1}^{L}$, where $c_l(\cdot)$ denotes the auxiliary after the l-th stage. Previous deeply-supervised nets [27] inspire this practice.

However, appending simply linear auxiliary classifiers without extra feature extraction modules may not be helpful to explore meaningful information, as empirically verified by [56]. Therefore, we try to design more complex auxiliary branches to provide informative task-aware knowledge as intended. Specifically, each auxiliary branch consists of a feature extraction module, a global average pooling layer and a linear classifier with a desirable dimension for the used auxiliary task, *e.g.* $N*M$ for our self-supervision augmented task. The feature extraction module is composed of several building blocks as same as the original backbone network, *i.e.* residual block in ResNets [1]. As empirically shown in [56], early layers may lack coarse features that encapsulate high-level semantic information, which is crucial for image-level recognition. To ensure the appropriate fine-to-coarse feature transformation, we make the path from an input to the end of an auxiliary branch with the same downsampling number as the backbone network. In Table I, we systematically show the overall architectural details of a ResNet-56 [1] as the backbone $f(\cdot)$ with its three auxiliary branches $\{c_l(\cdot)\}_{l=1}^{3}$ as an example for instantiating our design principle. Fig. 3a further illustrates the overall architecture schematically.

Formally, we describe the inference graph of the backbone network $f(\cdot)$ and its L attached auxiliary branches $\{c_l(\cdot)\}_{l=1}^{L}$. Given a transformed input sample $t_j(\boldsymbol{x})$, where $j \in \mathcal{M}$ and $t_1(\boldsymbol{x}) = \boldsymbol{x}$, the network $f(t_j(\boldsymbol{x}))$ outputs L intermediate feature-maps $\{\boldsymbol{F}_{l,j}\}_{l=1}^{L}$ from L convolutional stages respectively and a final primary class probability distribution $\boldsymbol{p}(t_j(\boldsymbol{x});\tau) = \sigma(f(t_j(\boldsymbol{x}))/\tau) \in \mathbb{R}^N$ over the original class space $\mathcal{N}$. We further feed $\{\boldsymbol{F}_{l,j}\}_{l=1}^{L}$ to the corresponding L auxiliary branches $\{c_l(\cdot)\}_{l=1}^{L}$ to learn desirable auxiliary tasks, respectively. Given the $\boldsymbol{F}_{l,j}$, the l-th branch $c_l(\cdot)$ can output a type of distribution from a specific auxiliary task, which is formulated as follows:

- a supervised class probability distribution

$$\boldsymbol{p}_l(\boldsymbol{x};\tau) = \sigma(c_l(\boldsymbol{F}_{l,j})/\tau) \in \mathbb{R}^N, s.t.\ j = 1, \quad (4)$$

  over the class space $\mathcal{N}$;
- a self-supervised class probability distribution

$$\boldsymbol{\mu}_l(t_j(\boldsymbol{x});\tau) = \sigma(c_l(\boldsymbol{F}_{l,j})/\tau) \in \mathbb{R}^M, s.t.\ 1 \leqslant j \leqslant M, \quad (5)$$

  over the class space $\mathcal{M}$;
- a self-supervision augmented distribution

$$\boldsymbol{q}_l(t_j(\boldsymbol{x});\tau) = \sigma(c_l(\boldsymbol{F}_{l,j})/\tau) \in \mathbb{R}^{N*M}, s.t.\ 1 \leqslant j \leqslant M, \quad (6)$$

  over the joint class space $\mathcal{N} \otimes \mathcal{M}$.

As referred above, we can change the dimension of the linear classifier in the l-th auxiliary branch c_l to deal with various classification tasks.

C. Training a Single Network with Auxiliary Branches

Given a single network backbone $f(\cdot)$ and its attached several auxiliary branches $\{c_l(\cdot)\}_{l=1}^{L}$, we train them jointly in an end-to-end manner, as shown in Fig. 3a. Given an input sample $\boldsymbol{x}$, we guide the $f(\cdot)$ to fit the ground-truth label $y \in \mathcal{N}$ using cross-entropy as the primary task loss:

$$\mathcal{L}_{task} = \mathcal{L}_{ce}(\boldsymbol{p}(\boldsymbol{x};\tau), y). \quad (7)$$

Where $\mathcal{L}_{ce}$ denotes the cross-entropy loss and $\boldsymbol{p}(\boldsymbol{x};\tau)$ is the supervised class probability distribution from f, *i.e.*, $\boldsymbol{p}(\boldsymbol{x};\tau) = \sigma(f(\boldsymbol{x})/\tau) \in \mathbb{R}^N$. This loss aims to make f fit the normal data for learning general classification capability and task-specific features.

For auxiliary branches, we can guide them to learn additional tasks, as referred in Section III-A. As for all L auxiliary branches, the losses of four tasks are respectively formulated as follow:

- *N-way supervised classification task*. This task is identical to the primary task. The loss for training L auxiliary branches is formulated as:

$$\mathcal{L}_{aux\_SCPD} = \sum_{l=1}^{L} \mathcal{L}_{ce}(\boldsymbol{p}_l(\boldsymbol{x};\tau), y). \tag{8}$$

  Here, $\boldsymbol{p}_l(\boldsymbol{x};\tau) \in \mathbb{R}^N$ is the predictive *Supervised Class Probability Distributions* (SCPD) from the l-th auxiliary branch.
- *M-way self-supervised classification task*. We can apply M transformations $\{t_j(\cdot)\}_{j=1}^{M}$ to the input sample $\boldsymbol{x}$ to construct M synthetic samples $\{t_j(\boldsymbol{x})\}_{j=1}^{M}$. We expect that the classifier can correctly recognize which transformation is applied to the synthetic sample. The loss for training L auxiliary branches is formulated as:

$$\mathcal{L}_{aux\_SSCPD} = \frac{1}{M}\sum_{j=1}^{M}\sum_{l=1}^{L} \mathcal{L}_{ce}(\boldsymbol{\mu}_l(t_j(\boldsymbol{x});\tau), j). \tag{9}$$

  Here, $\boldsymbol{\mu}_l(t_j(\boldsymbol{x});\tau) \in \mathbb{R}^M$ is the predictive *Self-Supervised Class Probability Distributions* (SSCPD) from the l-th auxiliary classifier. $j \in \mathcal{M}$ is the ground-truth label for the transformed sample $t_j(\boldsymbol{x})$. Inspired by [52], this loss would enable the network to learn self-supervised feature representations.
- *Multi-task classification task*. We can use two separated classification heads in each auxiliary branch to perform N-way supervised task and M-way self-supervised task, respectively. The loss for training L auxiliary branches is formulated as:

$$\begin{aligned}\mathcal{L}_{aux\_multi\_task} &= \mathcal{L}_{aux\_SCPD} + \mathcal{L}_{aux\_SSCPD} \\ &= \sum_{l=1}^{L} \mathcal{L}_{ce}(\boldsymbol{p}_l(\boldsymbol{x};\tau), y) \\ &\quad + \frac{1}{M}\sum_{j=1}^{M}\sum_{l=1}^{L} \mathcal{L}_{ce}(\boldsymbol{\mu}_l(t_j(\boldsymbol{x});\tau), j).\end{aligned} \tag{10}$$

- *$(N*M)$-way self-supervision augmented task*. We guide all auxiliary branches $\{c_l(\cdot)\}_{l=1}^{L}$ to learn self-supervision augmented tasks using cross-entropy to fit the joint ground-truth label $(y, j) \in \mathcal{N} \otimes \mathcal{M}$. The loss for training L auxiliary branches is formulated as:

$$\mathcal{L}_{aux\_SSAD} = \frac{1}{M}\sum_{j=1}^{M}\sum_{l=1}^{L} \mathcal{L}_{ce}(\boldsymbol{q}_l(t_j(\boldsymbol{x});\tau), (y, j)). \tag{11}$$

  Here, $\boldsymbol{q}_l(t_j(\boldsymbol{x});\tau) \in \mathbb{R}^{N*M}$ is the predictive *Self-Supervised Augmented Distributions* (SSAD) from the l-th auxiliary classifier. This loss forces auxiliary classifiers to solve the unified recognition of the categories of objects and transformations. It can implicitly enable the network to learn more informative and meaningful feature representations by dealing with a more competitive classification task.

We jointly train the backbone $f(\cdot)$ and L auxiliary branches $\{c_l(\cdot)\}_{l=1}^{L}$ by summarizing the primary task loss $\mathcal{L}_{task}$ and an auxiliary loss $\mathcal{L}_{aux}$ as:

$$\mathcal{L}^{single} = \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}(\mathcal{L}_{task} + \mathcal{L}_{aux}). \tag{12}$$

Here, $\mathcal{X}$ is the training set and $\mathcal{L}_{aux}$ can be any loss from $\{\mathcal{L}_{aux\_SCPD}, \mathcal{L}_{aux\_SSCPD}, \mathcal{L}_{aux\_multi\_task}, \mathcal{L}_{aux\_SSAD}\}$. During the test stage, we only retain the backbone network $f(\cdot)$ and discard L auxiliary branches $\{c_l(\cdot)\}_{l=1}^{L}$, leading to no extra inference costs compared with the original baseline network.

As validated in Section IV-B1, our proposed auxiliary loss $\mathcal{L}_{aux\_SSAD}$ can significantly improve the primarily supervised classification accuracy of the backbone network and surpass other alternative auxiliary losses by large margins. The results further inspire us to introduce $(N*M)$-way self-supervision augmented distribution as promising knowledge to the field of KD.

D. Teacher-Student Based Offline HSSAKD

The conventional KD aims to train a student from scratch by distilling meaningful knowledge from a pre-trained teacher network. Similar to the consistent protocol, we formulate our proposed offline HSSAKD by introducing a teacher network $f^T(\cdot) \cup \{c_l^T(\cdot)\}_{l=1}^{L}$ and a student network $f^S(\cdot) \cup \{c_l^S(\cdot)\}_{l=1}^{L}$. We conduct a two-stage pipeline: (1) training the teacher network at first; (2) training the student network under the supervision of the pre-trained frozen teacher network. Fig. 3b illustrates the overview schematically.

1) Training the Teacher Network: We denote the $\boldsymbol{p}^T(\boldsymbol{x};\tau) \in \mathbb{R}^N$ as the predictive class probaility distribution generated from $f^T(\cdot)$ and self-supervised augmented distributions $\{\boldsymbol{q}_l^T(t_j(\boldsymbol{x});\tau)\}_{l=1}^{L}$ generated from $\{c_l^T(\cdot)\}_{l=1}^{L}$ respectively. We train the teacher network using the primary task loss and an auxiliary loss referred in Section III-C, which is formulated as $\mathcal{L}^T$:

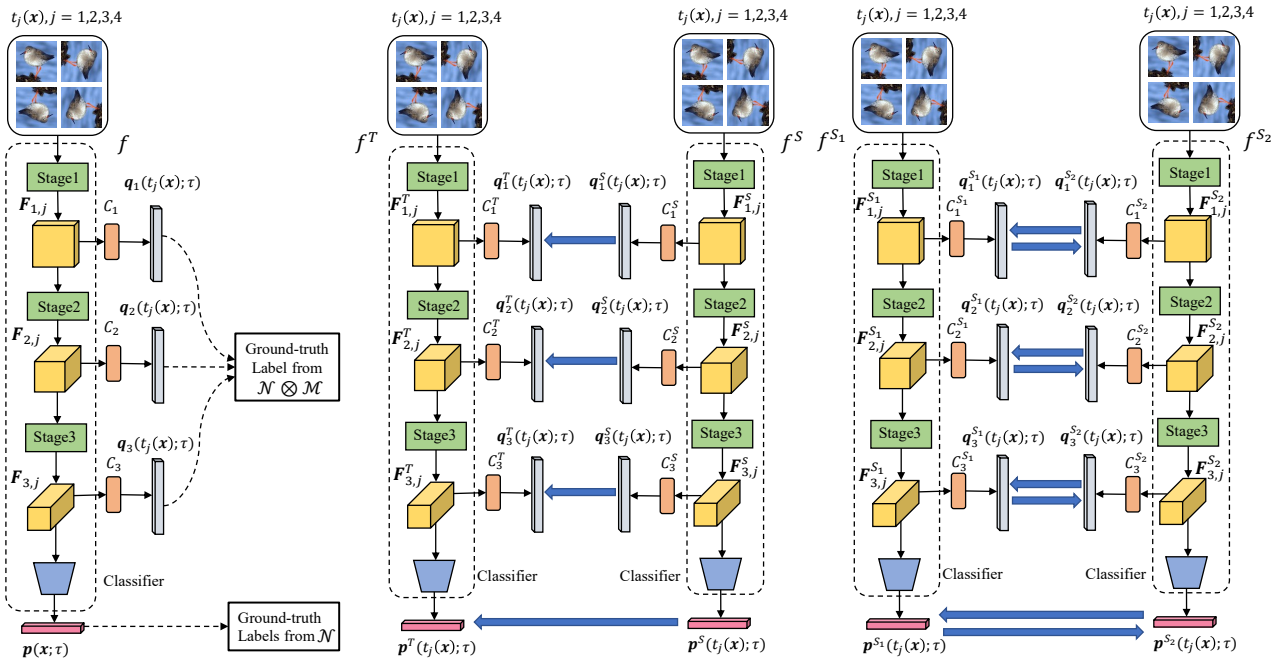
$$\begin{aligned}\mathcal{L}^T &= \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}[\mathcal{L}_{task}^T + \mathcal{L}_{aux\_SSAD}^T] \\ &= \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}[\mathcal{L}_{ce}(\boldsymbol{p}^T(\boldsymbol{x};\tau), y) \\ &\qquad + \frac{1}{M}\sum_{j=1}^{M}\sum_{l=1}^{L} \mathcal{L}_{ce}(\boldsymbol{q}_l^T(t_j(\boldsymbol{x});\tau), (y, j))].\end{aligned} \tag{13}$$

2) Training the Student Network: We conduct an end-to-end training process under the supervision of the teacher network. The overall loss includes a primary task loss from pre-defined ground-truth labels and mimicry losses from the pre-trained teacher network.

a) Task loss: Given an input sample $\boldsymbol{x}$, we train the student backbone $f^S(\cdot)$ to fit the ground-truth label $y \in \mathcal{N}$ using the cross-entropy loss:

$$\mathcal{L}_{task}^S = \mathcal{L}_{ce}(\boldsymbol{p}^S(\boldsymbol{x};\tau), y). \tag{14}$$

Where $\boldsymbol{p}^S(\boldsymbol{x};\tau)$ is the predictive class probability distribution generated from the $f^S(\cdot)$.



(a) Training a single network. (b) Teacher-student based offline HSSAKD. (c) Multi-student based online HSSAKD.

Fig. 3: **Overview of using our proposed self-supervision augmented distributions under three scenarios**. The blue arrow denotes the mimicry direction between two probability distributions by KL-divergence.

b) Mimicry loss from auxiliary self-supervision augmented distributions: We consider forcing hierarchical self-supervision augmented distributions $\{\boldsymbol{q}_l^S(t_j(\boldsymbol{x});\tau)\}_{l=1}^L$ from L auxiliary branches of the student network to mimic corresponding $\{\boldsymbol{q}_l^T(t_j(\boldsymbol{x});\tau)\}_{l=1}^L$ generated from L auxiliary branches of the teacher network, respectively. This distillation process performs in a one-to-one manner by KL-divergence function D_{KL}. The loss is formulated as Eq. (15), where the weight coefficient τ^2 is used to retain the gradient contributions unchanged as the temperature τ varies according to the KD recommendations [12].

$$\mathcal{L}_{KL\_q}^S = \frac{1}{M}\sum_{j=1}^{M}\sum_{l=1}^{L}\tau^2 D_{\mathrm{KL}}(\boldsymbol{q}_l^T(t_j(\boldsymbol{x});\tau) \parallel \boldsymbol{q}_l^S(t_j(\boldsymbol{x});\tau)). \tag{15}$$

Benefiting from Eq. (15), one can expect that the student network gains comprehensive guidances of unified self-supervised knowledge and the original supervised knowledge. The informative self-supervision augmented knowledge is derived from multi-scale intermediate feature maps encapsulated in hidden layers of the high-capacity teacher network. It leads to better representation learning for various-stage features of the student network.

c) Mimicry loss from the primary supervised class probability distribution: Similar to the conventional KD, we also transfer the supervised class probability distributions generated from the final layer between teacher $f^T(\cdot)$ and student $f^S(\cdot)$. Specifically, we transfer the class posterior derived from both the normal and transformed data denoted as $\{t_j(\boldsymbol{x})\}_{j=1}^M$, where $t_1(\boldsymbol{x}) = \boldsymbol{x}$. This loss is formulated as Eq. (16).

$$\mathcal{L}_{KL\_p}^S = \frac{1}{M}\sum_{j=1}^{M}\tau^2 D_{\mathrm{KL}}(\boldsymbol{p}^T(t_j(\boldsymbol{x});\tau) \parallel \boldsymbol{p}^S(t_j(\boldsymbol{x});\tau)). \tag{16}$$

It can be observed that we do not explicitly force the student backbone $f^S(\cdot)$ to fit the transformed data in task loss for preserving the normal classification capability. But we argue that mimicking the side product of $\{\boldsymbol{p}^T(t_j(\boldsymbol{x})\}_{j=2}^M$ from extra transformed data inferred from the teacher network is also beneficial for the feature representation learning of the student network. As validated in Section V-A, Eq. (16) can result in better performance than only distilling $\boldsymbol{p}^T(t_1(\boldsymbol{x});\tau)$ as same as the original KD.

d) Overall loss: We summarize the task loss and mimicry loss as the overall loss $\mathcal{L}_{KD}^S$ for training the student network.

$$\mathcal{L}_{KD}^S = \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}}(\mathcal{L}_{task}^S + \mathcal{L}_{KL\_q}^S + \mathcal{L}_{KL\_p}^S). \tag{17}$$

Following the broad practice, we set the hyper-parameter $\tau = 1$ in cross-entropy based task loss and $\tau = 3$ in KL-based mimicry loss, and we do not introduce other coefficients about loss weights. Empirically, we found this practice sufficient to work good enough.

3) Testing the Student Network: During the test stage, we only retain the student backbone network $f^S(\cdot)$ and discard L auxiliary branches $\{c_l^S(\cdot)\}_{l=1}^L$ as well as the teacher network $f^T(\cdot) \cup \{c_l^T(\cdot)\}_{l=1}^L$, leading to no extra inference costs compared with the original baseline network.

E. Multi-Student Based Online HSSAKD

Unlike the teacher-student based KD, online KD does not need a pre-trained teacher model as a prerequisite. Instead, it

aims to train a cohort of student models collaboratively from scratch. During the training stage, multiple students teach each other and perform knowledge transfer in an online manner. Therefore, online KD is an end-to-end training process rather than a two-stage pipeline in the conventional KD. We demonstrate that our offline HSSAKD can be extended to online HSSAKD. Fig. 3c illustrates the overview schematically.

In this section, we formulate the training graph as $K(K \geqslant 2)$ peer networks denoted as $\{f^{S_k}(\cdot) \cup \{c_l^{S_k}(\cdot)\}_{l=1}^{L}\}_{k=1}^{K}$, where the k-th network consists of a backbone $f^{S_k}(\cdot)$ and L auxiliary branches $\{c_l^{S_k}(\cdot)\}_{l=1}^{L}$. For each network, the training loss includes a primary task loss, an auxiliary task loss and mimicry losses.

1) Primary and auxiliary task losses: For the k-th network, we denote the $\boldsymbol{p}^{S_k}(\boldsymbol{x};\tau) \in \mathbb{R}^N$ as the primary supervised class probaility distribution generated from $f^{S_k}(\cdot)$ and L auxiliary self-supervision augmented distributions $\{\boldsymbol{q}_l^{S_k}(t_j(\boldsymbol{x});\tau)\}_{l=1}^{L}$ generated from $\{c_l^{S_k}(\cdot)\}_{l=1}^{L}$ respectively. For each network, we apply the same task loss and auxiliary loss referred in Section III-C. The total loss for training M networks are reformulated as Eq. (18) and Eq. (19), respectively:

$$\mathcal{L}_{task}^{S_1 \sim S_K} = \sum_{k=1}^{K} \mathcal{L}_{ce}(\boldsymbol{p}^{S_k}(\boldsymbol{x};\tau), y), \tag{18}$$

$$\mathcal{L}_{aux\_SSAD}^{S_1 \sim S_K} = \frac{1}{M} \sum_{k=1}^{K} \sum_{j=1}^{M} \sum_{l=1}^{L} \mathcal{L}_{ce}(\boldsymbol{q}_l^{S_k}(t_j(\boldsymbol{x});\tau), (y,j)). \tag{19}$$

2) Mutual mimicry loss between peer auxiliary self-supervision augmented distributions: DML [39] indicates that a cohort of student networks can benefit from transferring class probability distributions mutually in an online manner. Such a peer-teaching method makes a network learn better than training alone using the conventional supervised learning scenario. Inspired by this basic idea, we consider mutually mimicking our proposed self-supervision augmented distributions for online KD. Unlike the conventional DML that performs online KD only in the final layers, our HSSAKD considers more comprehensive knowledge derived from hierarchical feature maps encoded in hidden layers.

Taking two student networks f^{S_a} and f^{S_b} $(a \neq b,\ a,b \in [1,K])$ for example, we guide hierarchical self-supervision augmented distributions $\{\boldsymbol{q}_l^{S_a}(t_j(\boldsymbol{x});\tau)\}_{l=1}^{L}$ from L auxiliary branches of the S_a network to mimic corresponding $\{\boldsymbol{q}_l^{S_b}(t_j(\boldsymbol{x});\tau)\}_{l=1}^{L}$ generated from L auxiliary branches of the S_b network respectively, as shown in Eq. (20):

$$\mathcal{L}_{KL\_q}^{S_a \to S_b} = \frac{1}{M} \sum_{j=1}^{M} \sum_{l=1}^{L} \tau^2 D_{\mathrm{KL}}(\boldsymbol{q}_l^{S_b}(t_j(\boldsymbol{x});\tau) \parallel \boldsymbol{q}_l^{S_a}(t_j(\boldsymbol{x});\tau)), \tag{20}$$

and vice versa in Eq. (21):

$$\mathcal{L}_{KL\_q}^{S_b \to S_a} = \frac{1}{M} \sum_{j=1}^{M} \sum_{l=1}^{L} \tau^2 D_{\mathrm{KL}}(\boldsymbol{q}_l^{S_a}(t_j(\boldsymbol{x});\tau) \parallel \boldsymbol{q}_l^{S_b}(t_j(\boldsymbol{x});\tau)) \tag{21}$$

This bidirectional distillation process performs in a one-to-one manner by KL-divergence function D_{KL}.

As suggested by *Miyato et al.* [57], the gradient is not propagated through soft labels to avoid the model collapse problem. Therefore, we stop the gradient propagation of $\partial \mathcal{L}_{KL\_q}^{S_a \to S_b} / \partial \boldsymbol{q}_l^{S_b}(t_j(\boldsymbol{x});\tau)$ in Eq. (20) and $\partial \mathcal{L}_{KL\_q}^{S_b \to S_a} / \partial \boldsymbol{q}_l^{S_a}(t_j(\boldsymbol{x});\tau)$ in Eq. (21).

When $K \geqslant 2$, we perform mutual mimicry every two of K student networks to learn more diverse knowledge captured by other networks, leading to the total loss as:

$$\mathcal{L}_{KL\_q}^{S_1 \sim S_K} = \sum_{1 \leqslant a,b \leqslant K} (\mathcal{L}_{KL\_q}^{S_a \to S_b} + \mathcal{L}_{KL\_q}^{S_b \to S_a}). \tag{22}$$

3) Mutual mimicry loss between primary supervised class probability distributions: We also conduct mutual mimicry of supervised class probability distributions generated from the final layers among multiple student networks. These distributions are also derived from both the normal and transformed data denoted as $\{t_j(\boldsymbol{x})\}_{j=1}^{M}$. For two peer networks f^{S_a} and f^{S_b} $(a \neq b,\ a,b \in [1,K])$, we guide $\{\boldsymbol{p}^{S_a}(t_j(\boldsymbol{x};\tau)\}_{j=1}^{M}$ generated from f^{S_a} to mimic $\{\boldsymbol{p}^{S_b}(t_j(\boldsymbol{x};\tau)\}_{j=1}^{M}$ generated from f^{S_b}, as shown in Eq. (23):

$$\mathcal{L}_{KL\_p}^{S_a \to S_b} = \frac{1}{M} \sum_{j=1}^{M} \tau^2 D_{\mathrm{KL}}(\boldsymbol{p}^{S_b}(t_j(\boldsymbol{x});\tau) \parallel \boldsymbol{p}^{S_a}(t_j(\boldsymbol{x});\tau)), \tag{23}$$

and vice versa in Eq. (24):

$$\mathcal{L}_{KL\_p}^{S_b \to S_a} = \frac{1}{M} \sum_{j=1}^{M} \tau^2 D_{\mathrm{KL}}(\boldsymbol{p}^{S_a}(t_j(\boldsymbol{x});\tau) \parallel \boldsymbol{p}^{S_b}(t_j(\boldsymbol{x});\tau)). \tag{24}$$

Here, we stop the gradient propagation of $\partial \mathcal{L}_{KL\_p}^{S_a \to S_b} / \partial \boldsymbol{p}^{S_b}(t_j(\boldsymbol{x});\tau)$ in Eq. (23) and $\partial \mathcal{L}_{KL\_q}^{S_b \to S_a} / \partial \boldsymbol{p}^{S_b}(t_j(\boldsymbol{x});\tau)$ in Eq. (24). When $K \geqslant 2$, we perform mutual mimicry every two of K student networks, leading to the total loss as:

$$\mathcal{L}_{KL\_p}^{S_1 \sim S_K} = \sum_{1 \leqslant a,b \leqslant K} (\mathcal{L}_{KL\_p}^{S_a \to S_b} + \mathcal{L}_{KL\_p}^{S_b \to S_a}). \tag{25}$$

4) Overall loss: We summarize the above three loss to perform online KD over K student networks:

$$\mathcal{L}_{OKD}^{S_1 \sim S_K} = \mathbb{E}_{\boldsymbol{x} \in \mathcal{X}}(\mathcal{L}_{task}^{S_1 \sim S_K} + \mathcal{L}_{aux\_SSAD}^{S_1 \sim S_K} + \mathcal{L}_{KL\_q}^{S_1 \sim S_K} + \mathcal{L}_{KL\_p}^{S_1 \sim S_K}). \tag{26}$$

As suggested by DML [39], we set the hyper-parameter $\tau = 1$ in cross-entropy based task loss and $\tau = 3$ in KL-based mimicry loss, and we do not introduce other coefficients about loss weights. Empirically, we found this practice sufficient to work good enough.

5) Testing the Student Network: During the test stage, we can retain any student backbone network from $\{f^{S_k}(\cdot)\}_{k=1}^{K}$ and discard all auxiliary branches $\{\{c_l^{S_k}(\cdot)\}_{l=1}^{L}\}_{k=1}^{K}$, leading to no extra inference costs compared with the original baseline network.

IV. Experiments

A. Datasets and Experimental setup

a) Image Classification on CIFAR-100: CIFAR-100 [60] is composed of 50K training images and 10K test images in

TABLE II: Accuracy (%) of our method for training a single network with auxiliary branches on CIFAR-100. We use four different tasks referred in Section III-A to train auxiliary branches. During the test stage, we discard all auxiliary branches and only retain the backbone network, leading to no extra inference costs compared with the original baseline network. The numbers in **Bold** denote the best results. 'Gain(%)' denotes the performance improvement of our method over baseline.

Network	Params	Baseline Eq. (7)	Train a single network with auxiliary branches				Gain(%)↑
			+Supervised Eq. (8)	+Self-supervised Eq. (9)	+Multi-task Eq. (10)	+Self-supervision augmented Eq. (11)	
WRN-40-2 [58]	2.26M	$76.44_{(\pm 0.20)}$	$77.00_{(\pm 0.23)}$	$78.35_{(\pm 0.20)}$	$78.53_{(\pm 0.23)}$	$\mathbf{80.90}_{(\pm 0.22)}$	4.46
WRN-40-1 [58]	0.57M	$71.95_{(\pm 0.59)}$	$72.36_{(\pm 0.12)}$	$73.58_{(\pm 0.25)}$	$73.38_{(\pm 0.28)}$	$\mathbf{75.73}_{(\pm 0.10)}$	3.78
ResNet-56 [1]	0.86M	$73.00_{(\pm 0.17)}$	$73.61_{(\pm 0.14)}$	$74.82_{(\pm 0.23)}$	$74.60_{(\pm 0.26)}$	$\mathbf{77.52}_{(\pm 0.43)}$	4.52
ResNet-32×4 [1]	7.43M	$79.56_{(\pm 0.23)}$	$80.09_{(\pm 0.21)}$	$80.53_{(\pm 0.37)}$	$80.57_{(\pm 0.18)}$	$\mathbf{83.58}_{(\pm 0.16)}$	4.02
VGG-13 [59]	9.46M	$75.35_{(\pm 0.21)}$	$75.56_{(\pm 0.20)}$	$76.02_{(\pm 0.34)}$	$76.36_{(\pm 0.23)}$	$\mathbf{78.56}_{(\pm 0.09)}$	3.21
MobileNetV2 [6]	0.81M	$73.51_{(\pm 0.26)}$	$74.12_{(\pm 0.31)}$	$74.87_{(\pm 0.18)}$	$75.16_{(\pm 0.19)}$	$\mathbf{77.39}_{(\pm 0.18)}$	3.88
ShuffleNetV1 [5]	0.95M	$71.74_{(\pm 0.35)}$	$72.41_{(\pm 0.28)}$	$73.21_{(\pm 0.32)}$	$73.75_{(\pm 0.25)}$	$\mathbf{76.19}_{(\pm 0.14)}$	4.45
ShuffleNetV2 [7]	1.36M	$72.96_{(\pm 0.33)}$	$73.59_{(\pm 0.17)}$	$74.50_{(\pm 0.33)}$	$74.93_{(\pm 0.12)}$	$\mathbf{77.82}_{(\pm 0.13)}$	4.86

100 classes. We use the standard data augmentation following [1]. We train all networks by the SGD optimizer with a momentum of 0.9, a batch size of 64 and a weight decay of 5×10^{-4}. The initial learning rate starts at 0.05 and is decayed by a factor of 10 at 150, 180 and 210 epochs within the total 240 epochs. For light-weight networks of MobileNetV2, ShuffleNetV1 and ShuffleNetV2, we use an initial learning rate of 0.01.

b) Image Classification on ImageNet: ImageNet [61] is a large-scale image classification dataset composed of 1.2 million training images and 50K validation images in 1000 classes. We use the standard data augmentation following [1]. We train all networks by the SGD optimizer with a momentum of 0.9, a batch size of 256 and a weight decay of 1×10^{-4}. The initial learning rate starts at 0.1 and is decayed by a factor of 10 at the 30-th, 60-th and 90-th epochs within the total 100 epochs.

c) Transfer Experiments of Image Classification on STL-10 and TinyImageNet: STL-10 [62] is composed of 5K labelled training images and 8K test images in 10 classes. TinyImageNet[1] is composed of 100K training images and 10k test images in 200 classes. For data augmentation, we randomly crop the image into the size ranging from $(0.08, 1.0)$ of the original size and a random aspect ratio ranging from $(3/4, 4/3)$ of the original aspect ratio. The cropped patch is resized to 32×32 and is randomly horizontal flipped with a probability of 0.5. We train the linear classifiers by the SGD optimizer with a momentum of 0.9, a batch size of 64 and a weight decay of 0. The initial learning rate starts at 0.1 and is decayed by a factor of 10 at the 30-th, 60-th and 90-th epochs within the total 100 epochs.

d) Transfer Experiments of Object Detection on Pascal VOC: Following the consistent protocol, we use Pascal VOC [63] `trainval07+12` for training and `test07` for evaluation. The result set consists of 16551 training images and 4952 test images in 20 classes. The image scale is 800×1000 pixels during training and inference. We train the model by SGD optimizer with a momentum of 0.9 and a weight decay of 1×10^{-4}. The initial learning rate starts at 0.001 and is decayed by a factor of 10 at the 3-th epoch within the total 4 epochs.

[1]http://tiny-imagenet.herokuapp.com/

e) Auxiliary self-supervised classification task: Unless otherwise specified, we employ 4-way rotations from $\{0°, 90°, 180°, 270°\}$ as an auxiliary self-supervised task, as illustrated in Fig. 3.

B. Experiments on CIFAR-100 and ImageNet Classification

1) Training a Single Network with Auxiliary Branches: Initially, we examine the efficacy of four different auxiliary tasks referred in Section III-A for enhancing the representation learning over a single network for solving the primary supervised task. As illustrated in Fig. 3a, we guide all auxiliary branches to learn additional tasks. The results about four different auxiliary tasks over the baseline are shown in Table II.

Although forcing auxiliary branches to learn various tasks can consistently improve performance over the original baseline network, using our proposed self-supervision augmented task as an auxiliary task can result in maximum accuracy gains. Moreover, we find that using a supervised task as the same as the primary task only leads to marginal improvements with an average gain of 0.53% upon the baseline across various networks. This is because the homogeneous supervised task may not contribute much extra knowledge. Somewhat surprisingly, using a self-supervised task that is orthogonal to the primary task can lead to moderate improvements with an average gain of 1.60%. This is possibly because this self-supervised task can bring extra self-supervision knowledge and explicitly facilitate feature representation learning. We further find that incorporating supervised and self-supervised task into a multi-task framework only achieves comparable performance with the self-supervised counterpart.

In contrast, our self-supervision augmented task leads to significant improvements with an average gain of 4.15%. The results suggest that the self-supervision augmented task can explore more meaningful joint knowledge to encourage better feature representations than any single task or multi-task learning.

2) Teacher-Student Based Offline HSSAKD: Based on the success of the self-supervision augmented task, we consider distilling self-supervision augmented distributions as new knowledge from hierarchical feature maps between teacher and student networks. As shown in Table III and Table IV, we compare our HSSAKD with other popular KD methods across

TABLE III: Top-1 accuracy (%) comparison of SOTA distillation methods across various teacher-student pairs on CIFAR-100. *All results with the form $mean_{(\pm std)}$ are implemented by ourselves with three runs.* The numbers in **Bold** and underline denote the best and the second-best results, respectively. 'Teacher' denotes that we first train the backbone $f^T(\cdot)$ and then train auxiliary classifiers $\{c_l^T(\cdot)\}_{l=1}^L$ based on the frozen $f^T(\cdot)$. For a fair comparison, all compared methods and 'Ours' are supervised by 'Teacher'. 'Teacher*' denotes that we train $f^T(\cdot)$ and $\{c_l^T(\cdot)\}_{l=1}^L$ jointly, leading to a more powerful teacher network. 'Ours*' denotes the results supervised by 'Teacher*' for pursuing better performance.

Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	ResNet56 ResNet20	ResNet32×4 ResNet8×4	VGG13 MobileNetV2	ResNet50 MobileNetV2	WRN-40-2 ShuffleNetV1	ResNet32×4 ShuffleNetV2
Params	2.26M 0.70M	2.26M 0.57M	0.86M 0.28M	7.43M 1.23M	9.46M 0.81M	23.71M 0.81M	2.26M 0.95M	7.43M 1.36M
Teacher	76.45	76.45	73.44	79.63	74.64	79.34	76.45	79.63
Teacher*	80.70	80.70	77.20	83.73	78.48	83.85	80.70	83.73
Student	$73.57_{(\pm0.23)}$	$71.95_{(\pm0.59)}$	$69.62_{(\pm0.26)}$	$72.95_{(\pm0.24)}$	$73.51_{(\pm0.26)}$	$73.51_{(\pm0.26)}$	$71.74_{(\pm0.35)}$	$72.96_{(\pm0.33)}$
KD [12]	$75.23_{(\pm0.23)}$	$73.90_{(\pm0.44)}$	$70.91_{(\pm0.10)}$	$73.54_{(\pm0.26)}$	$75.21_{(\pm0.24)}$	$75.80_{(\pm0.46)}$	$75.83_{(\pm0.18)}$	$75.43_{(\pm0.33)}$
FitNet [18]	$75.30_{(\pm0.42)}$	$74.30_{(\pm0.42)}$	$71.21_{(\pm0.16)}$	$75.37_{(\pm0.12)}$	$75.42_{(\pm0.34)}$	$75.41_{(\pm0.07)}$	$76.27_{(\pm0.18)}$	$76.91_{(\pm0.06)}$
AT [19]	$75.64_{(\pm0.31)}$	$74.32_{(\pm0.23)}$	$71.35_{(\pm0.09)}$	$75.06_{(\pm0.19)}$	$74.08_{(\pm0.21)}$	$76.57_{(\pm0.20)}$	$76.51_{(\pm0.44)}$	$76.32_{(\pm0.12)}$
AB [20]	$71.26_{(\pm1.32)}$	$74.55_{(\pm0.46)}$	$71.56_{(\pm0.19)}$	$74.31_{(\pm0.09)}$	$74.98_{(\pm0.44)}$	$75.87_{(\pm0.39)}$	$76.43_{(\pm0.09)}$	$76.40_{(\pm0.29)}$
VID [21]	$75.31_{(\pm0.22)}$	$74.23_{(\pm0.28)}$	$71.35_{(\pm0.09)}$	$75.07_{(\pm0.35)}$	$75.67_{(\pm0.13)}$	$75.97_{(\pm0.08)}$	$76.24_{(\pm0.44)}$	$75.98_{(\pm0.41)}$
RKD [31]	$75.33_{(\pm0.14)}$	$73.90_{(\pm0.26)}$	$71.67_{(\pm0.08)}$	$74.17_{(\pm0.22)}$	$75.54_{(\pm0.36)}$	$76.20_{(\pm0.06)}$	$75.74_{(\pm0.32)}$	$75.42_{(\pm0.25)}$
SP [32]	$74.35_{(\pm0.59)}$	$72.91_{(\pm0.24)}$	$71.45_{(\pm0.38)}$	$75.44_{(\pm0.11)}$	$75.68_{(\pm0.35)}$	$76.35_{(\pm0.14)}$	$76.40_{(\pm0.37)}$	$76.43_{(\pm0.21)}$
CC [22]	$75.30_{(\pm0.03)}$	$74.46_{(\pm0.05)}$	$71.44_{(\pm0.10)}$	$74.40_{(\pm0.24)}$	$75.66_{(\pm0.33)}$	$76.05_{(\pm0.25)}$	$75.63_{(\pm0.30)}$	$75.74_{(\pm0.18)}$
CRD [23]	$75.81_{(\pm0.33)}$	$74.76_{(\pm0.25)}$	$71.83_{(\pm0.42)}$	$75.77_{(\pm0.24)}$	$76.13_{(\pm0.16)}$	$76.89_{(\pm0.27)}$	$76.37_{(\pm0.23)}$	$76.51_{(\pm0.09)}$
SSKD [28]	$\underline{76.16}_{(\pm0.17)}$	$\underline{75.84}_{(\pm0.04)}$	$70.80_{(\pm0.02)}$	$75.83_{(\pm0.29)}$	$76.21_{(\pm0.16)}$	$\underline{78.21}_{(\pm0.16)}$	$76.71_{(\pm0.31)}$	$77.64_{(\pm0.24)}$
TOFD [64]	$75.48_{(\pm0.21)}$	$74.45_{(\pm0.29)}$	$\underline{72.02}_{(\pm0.34)}$	$75.45_{(\pm0.02)}$	$76.05_{(\pm0.16)}$	$76.40_{(\pm0.13)}$	$75.80_{(\pm0.20)}$	$76.47_{(\pm0.22)}$
SemCKD [65]	$75.02_{(\pm0.16)}$	$73.54_{(\pm0.83)}$	$71.54_{(\pm0.11)}$	$\underline{75.89}_{(\pm0.05)}$	$\underline{76.45}_{(\pm0.28)}$	$77.23_{(\pm0.44)}$	$\underline{77.39}_{(\pm0.35)}$	$\underline{78.26}_{(\pm0.24)}$
HSSAKD (Ours)	$77.20_{(\pm0.17)}$	$77.00_{(\pm0.21)}$	$72.58_{(\pm0.33)}$	$77.26_{(\pm0.14)}$	$77.45_{(\pm0.21)}$	$78.79_{(\pm0.11)}$	$78.51_{(\pm0.20)}$	$79.93_{(\pm0.11)}$
HSSAKD (Ours*)	$\mathbf{78.67}_{(\pm0.20)}$	$\mathbf{78.12}_{(\pm0.25)}$	$\mathbf{73.73}_{(\pm0.10)}$	$\mathbf{77.69}_{(\pm0.05)}$	$\mathbf{79.27}_{(\pm0.12)}$	$\mathbf{79.43}_{(\pm0.24)}$	$\mathbf{80.11}_{(\pm0.32)}$	$\mathbf{80.86}_{(\pm0.15)}$

TABLE IV: Top-1 accuracy (%) comparison of SOTA offline KD methods over ResNet-34-ResNet-18 on ImageNet.

Acc	Teacher	Teacher*	Student	KD [12]	AT [19]	CC [22]	SP [32]	RKD [31]	CRD [23]	SSKD [28]	Ours	Ours*
Top-1	73.31	75.48	69.75	70.66	70.70	69.96	70.62	71.34	71.38	<u>71.62</u>	72.16	**72.39**
Top-5	91.42	92.67	89.07	89.88	90.00	89.17	89.80	90.37	90.49	<u>90.67</u>	90.85	**91.00**

TABLE V: Top-1 accuracy(%) comparisons of our method against SOTA online mutual knowledge distillation methods on CIFAR-100 and ImageNet. *All results with the form $mean_{(\pm std)}$ are implemented by ourselves with three runs.* The numbers in **Bold** and underline denote the best and the second-best results, respectively. 'Gain(%)' denotes the performance improvement of our method over the second-best result.

Network	Baseline	DML [39]	ONE [41]	OKDDip [42]	KDCL [43]	AFD [44]	HSSAKD (Ours)	Gain(%)↑
WRN-40-2	$76.44_{(\pm0.20)}$	$78.96_{(\pm0.13)}$	$78.76_{(\pm0.14)}$	$79.23_{(\pm0.34)}$	$78.50_{(\pm0.29)}$	$\underline{79.54}_{(\pm0.13)}$	$\mathbf{82.58}_{(\pm0.21)}$	3.04
WRN-40-1	$71.95_{(\pm0.59)}$	$74.33_{(\pm0.12)}$	$74.56_{(\pm0.25)}$	$74.89_{(\pm0.29)}$	$74.20_{(\pm0.22)}$	$\underline{75.23}_{(\pm0.05)}$	$\mathbf{76.67}_{(\pm0.41)}$	1.44
ResNet-56	$73.00_{(\pm0.17)}$	$75.40_{(\pm0.24)}$	$75.64_{(\pm0.12)}$	$76.25_{(\pm0.38)}$	$75.28_{(\pm0.13)}$	$\underline{76.78}_{(\pm0.28)}$	$\mathbf{78.16}_{(\pm0.56)}$	1.38
ResNet-32×4	$79.56_{(\pm0.23)}$	$81.61_{(\pm0.27)}$	$82.34_{(\pm0.19)}$	$81.21_{(\pm0.10)}$	$80.76_{(\pm0.07)}$	$\underline{82.91}_{(\pm0.17)}$	$\mathbf{84.91}_{(\pm0.19)}$	2.00
VGG-13	$75.35_{(\pm0.21)}$	$77.77_{(\pm0.19)}$	$78.45_{(\pm0.34)}$	$77.32_{(\pm0.09)}$	$77.22_{(\pm0.16)}$	$\underline{78.61}_{(\pm0.09)}$	$\mathbf{80.44}_{(\pm0.05)}$	1.83
MobileNetV2	$73.51_{(\pm0.26)}$	$76.57_{(\pm0.16)}$	$76.13_{(\pm0.17)}$	$76.79_{(\pm0.41)}$	$75.63_{(\pm0.20)}$	$\underline{77.05}_{(\pm0.37)}$	$\mathbf{78.85}_{(\pm0.13)}$	1.80
ShuffleNetV1	$71.74_{(\pm0.35)}$	$75.01_{(\pm0.42)}$	$74.63_{(\pm0.08)}$	$75.41_{(\pm0.29)}$	$74.82_{(\pm0.59)}$	$\underline{75.82}_{(\pm0.34)}$	$\mathbf{78.34}_{(\pm0.03)}$	2.52
ShuffleNetV2	$72.96_{(\pm0.33)}$	$76.41_{(\pm0.18)}$	$75.74_{(\pm0.20)}$	$76.29_{(\pm0.22)}$	$75.10_{(\pm0.35)}$	$\underline{77.21}_{(\pm0.14)}$	$\mathbf{79.98}_{(\pm0.12)}$	2.77

widely used network architectures on CIFAR-100 and ImageNet. Our HSSAKD achieves the best performance among all KD methods. It is noteworthy that HSSAKD can significantly outperform the previous SOTA method SSKD [28] by an average accuracy gain of 2.56% on CIFAR-100 and a top-1 gain of 0.77% on ImageNet. The results verify that encoding the self-supervised auxiliary task as an augmented distribution in our HASKD is better than self-supervised contrastive distributions in SSKD to perform KD-based representation learning.

3) Multi-Student Based Online HSSAKD: Based on the success in offline HSSAKD, it is also interesting to extend our HSSAKD framework to the scenario of online KD for mutually transferring self-supervision augmented distributions. As shown in Table V, we compare HSSAKD with other representative online KD methods across widely used network architectures on CIFAR-100 and ImageNet. For fair comparisons, all methods use two identical peer networks (*i.e.* $K = 2$) for online KD. Our HSSAKD achieves the best performance among all online KD methods. Specifically, HSSAKD can significantly outperforms the seminal DML [39] and previous SOTA method AFD [44] by an average accuracy gain of 2.98% and 2.20% on CIFAR-100, respectively. Further experiments on the large-scale ImageNet for collaboratively training two ResNet-18 verify the superiority of our HSSAKD. We re-

TABLE VI: Top-1 accuracy (%) comparison of SOTA online KD methods on ImageNet.

Network	Baseline	DML [39]	ONE [41]	OKDDip [42]	KDCL [43]	AFD [44]	HSSAKD (Ours)
ResNet-18	69.75	70.48	70.55	70.73	70.83	70.51	**71.49**

TABLE VII: Linear classification accuracy (%) of transfer learning on the student MobileNetV2 pre-trained using the teacher VGG-13. The numbers in **Bold** and underline denote the best and the second-best results, respectively.

Transferred Dataset	Teacher	Student	KD	FitNet	AT	AB	VID	RKD	SP	CC	CRD	SSKD	Ours
CIFAR-100→ STL-10	63.08	67.76	67.90	69.41	67.37	67.82	69.29	69.74	68.96	69.13	70.09	71.03	**74.66**
CIFAR-100→ TinyImageNet	29.53	34.69	34.15	36.04	34.44	34.79	36.09	37.21	35.69	36.43	38.17	39.07	**42.57**

TABLE VIII: Comparison of detection AP (%) (Average Precision) on individual classes and mAP (mean Average Precision) (%) on Pascal VOC [63] using ResNet-18 as the backbone pre-trained by various KD methods over the Faster-RCNN system [2]. The numbers in **Bold** and underline denote the best and the second-best results, respectively.

Method	mAP	AP (Average Precision)																			
		areo	bike	brid	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Baseline	76.2	79.0	83.4	77.5	63.4	65.0	80.3	85.5	86.7	59.4	81.7	69.1	84.2	84.2	81.3	83.7	48.3	80.0	73.6	82.6	74.6
KD [12]	77.1	78.6	84.4	78.0	68.0	62.8	82.9	85.9	**88.4**	60.7	81.5	68.6	85.9	84.4	82.7	84.5	48.8	79.5	74.8	**85.0**	75.9
CRD [23]	77.4	77.5	**84.9**	77.3	66.3	65.6	82.3	86.6	88.2	62.1	81.8	70.8	85.7	84.9	82.8	84.3	51.6	81.1	75.2	83.1	75.2
SSKD [28]	77.6	78.6	84.2	78.2	66.2	63.3	82.8	86.1	87.3	**63.4**	**84.4**	**71.8**	84.5	**85.1**	**83.1**	83.9	51.9	80.1	**77.2**	84.1	76.0
HSSAKD	**78.4**	**84.8**	**84.9**	**81.8**	**68.2**	**66.9**	**84.1**	**86.7**	88.2	62.1	82.6	70.3	**86.4**	84.3	82.3	**84.8**	**53.2**	**81.3**	74.8	84.2	**78.2**

mark that previous online KD methods, such as DML [39], ONE [41] and KDCL [43], often focus on distilling the conventional supervised class probability distributions and only differ in the mechanisms for distillation or producing an online teacher. In contrast, our HSSAKD goes on a step further to introduce a new type of knowledge of self-supervision augmented distributions, which achieves promising results in the field of online KD.

C. Transfer Experiments of Learned Feature Representations

1) Transfer Experiments on STL-10 and TinyImageNet for Image Classification: Beyond the accuracy on the object dataset, we also expect that the student network can produce the generalized feature representations that transfer well to other unseen semantic recognition datasets. To this end, we freeze the feature extractor pre-trained on the upstream CIFAR-100, and then train two linear classifiers based on frozen pooled features for downstream STL-10 and TinyImageNet respectively, following the common linear classification protocol [23]. As shown in Table VII, we can observe that both SSKD and HSSAKD achieve better accuracy than other comparative methods, demonstrating that using self-supervision auxiliary tasks for distillation is conducive to generating better feature representations. Moreover, HSSAKD can significantly outperform the best-competing SSKD by an accuracy gain of 3.63% on STL-10 and an accuracy gain of 3.50% on TinyImageNet.

a) Transfer Experiments on Pascal VOC for Object Detection: We further evaluate the student network ResNet-18 pre-trained with the teacher ResNet-34 on ImageNet as a backbone to carry out downstream object detection on Pascal VOC [63]. We use Faster-RCNN [2] framework and follow the standard data preprocessing and finetuning strategy. The comparison on detection performance towards AP (Average Precision) on individual classes and mAP (mean Average Precision) is shown in Table VIII. Our method outperforms the original baseline by 2.27% mAP and the best-competing SSKD by 0.85% mAP. Moreover, our method can also achieve the best or second-best AP in most classes compared with other widely used KD methods. These results verify that our method can guide a network to learn better feature representations for semantic recognition tasks.

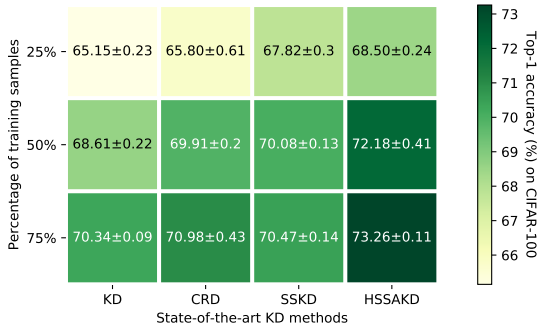


Fig. 4: Top-1 accuracy (%) comparison on CIFAR-100 under few-shot scenario with various percentages of training samples. We use the ResNet56-ResNet20 as the teacher-student pair for evaluation.

D. Efficacy under Few-shot Scenario

We compare our method with conventional KD and SOTA CRD and SSKD under few-shot scenarios by retaining 25%, 50% and 75% training samples. For a fair comparison, we use the same data split with a stratified sampling strategy for each few-shot setting while maintaining the original test set. As shown in Fig. 4, our method can consistently surpass other offline KD methods by large margins under various few-shot settings. Moreover, it is noteworthy that by using only 25% training samples, our method can achieve comparable accuracy with the baseline trained on the complete set. This is because

TABLE IX: Top-1 accuracy (%) comparison towards ablation study of loss terms on the student networks of WRN-16-2 and ShuffleNetV1 supervised through the pre-trained teacher network WRN-40-2 for *offline KD* on CIFAR-100.

Loss terms	WRN-16-2	ShuffleNetV1
$\mathcal{L}^S_{task}$	73.57$_{(\pm 0.23)}$	71.74$_{(\pm 0.35)}$
$\mathcal{L}^S_{task} + \mathcal{L}^S_{KL\_q}$	77.11$_{(\pm 0.13)}$	78.87$_{(\pm 0.25)}$
$\mathcal{L}^S_{task} + \mathcal{L}^S_{KL\_q}$+KD [12]	78.23$_{(\pm 0.23)}$	79.06$_{(\pm 0.30)}$
$\mathcal{L}^S_{task} + \mathcal{L}^S_{KL\_q} + \mathcal{L}^S_{KL\_p}$	**78.67**$_{(\pm 0.20)}$	**80.11**$_{(\pm 0.32)}$

TABLE X: Top-1 accuracy (%) comparison towards ablation study of loss terms on the student networks of WRN-16-2 and ShuffleNetV1 for *online mutual KD* on CIFAR-100.

Loss terms	WRN-16-2	ShuffleNetV1
$\mathcal{L}^{S_1\sim S_2}_{task}$	73.57$_{(\pm 0.23)}$	71.74$_{(\pm 0.35)}$
$\mathcal{L}^{S_1\sim S_2}_{task} + \mathcal{L}^{S_1\sim S_2}_{aux\_SSAD}$	76.15$_{(\pm 0.31)}$	76.19$_{(\pm 0.14)}$
$\mathcal{L}^{S_1\sim S_2}_{task} + \mathcal{L}^{S_1\sim S_2}_{aux\_SSAD} + \mathcal{L}^{S_1\sim S_2}_{KL\_q}$	76.83$_{(\pm 0.29)}$	77.57$_{(\pm 0.21)}$
$\mathcal{L}^{S_1\sim S_2}_{task} + \mathcal{L}^{S_1\sim S_2}_{aux\_SSAD} + \mathcal{L}^{S_1\sim S_2}_{KL\_q}$+DML [39]	76.89$_{(\pm 0.35)}$	77.73$_{(\pm 0.17)}$
$\mathcal{L}^{S_1\sim S_2}_{task} + \mathcal{L}^{S_1\sim S_2}_{aux\_SSAD} + \mathcal{L}^{S_1\sim S_2}_{KL\_q} + \mathcal{L}^{S_1\sim S_2}_{KL\_p}$	**77.30**$_{(\pm 0.15)}$	**78.34**$_{(\pm 0.03)}$

TABLE XI: Top-1 accuracy (%) comparison towards various distributions as knowledge for *offline KD* on the student networks of WRN-16-2 and ShuffleNetV1 supervised through the pre-trained teacher network WRN-40-2 on CIFAR-100.

Knowledge for KD	WRN-16-2	ShuffleNetV1
None	73.57$_{(\pm 0.23)}$	71.74$_{(\pm 0.35)}$
Supervised $\boldsymbol{p} \in \mathbb{R}^N$	74.73$_{(\pm 0.33)}$	73.24$_{(\pm 0.27)}$
Self-supervised $\boldsymbol{\mu} \in \mathbb{R}^M$	75.22$_{(\pm 0.13)}$	73.62$_{(\pm 0.15)}$
Multi-task $\boldsymbol{p} \in \mathbb{R}^N$ and $\boldsymbol{\mu} \in \mathbb{R}^M$	74.94$_{(\pm 0.18)}$	74.10$_{(\pm 0.26)}$
Self-supervision augmented $\boldsymbol{q} \in \mathbb{R}^{N*M}$	**78.67**$_{(\pm 0.20)}$	**80.11**$_{(\pm 0.32)}$

TABLE XII: Top-1 accuracy (%) comparison towards various self-supervised pretext tasks for *offline HSSAKD* on the student networks of WRN-16-2 and ShuffleNetV1 supervised through the pre-trained teacher network WRN-40-2 on CIFAR-100.

Self-supervised pretext task	WRN-16-2	ShuffleNetV1
6-way color channal permutation	75.28$_{(\pm 0.16)}$	77.05$_{(\pm 0.26)}$
4-way jigsaw puzzles	77.15$_{(\pm 0.24)}$	79.23$_{(\pm 0.35)}$
4-way random rotations	**78.67**$_{(\pm 0.20)}$	**80.11**$_{(\pm 0.32)}$

our method can effectively learn general feature representations from limited data. In contrast, previous methods often focus on mimicking the inductive biases from intermediate feature maps or cross-sample relationships, which may overfit the limited set and generalize worse to the test set.

V. Ablation Study and Analysis

A. Effect of loss terms

To examine each component of our proposed HSSAKD, we compare three variants under the conventional offline and online scenarios settings, respectively. As shown in Table IX, for offline HSSAKD, three variants are (1) our core loss $\mathcal{L}^S_{KL\_q}$ for distilling self-supervision augmented distributions; (2) $\mathcal{L}^S_{KL\_q}$ +conventional KD [12]; (3) $\mathcal{L}^S_{KL\_q} + \mathcal{L}^S_{KL\_p}$. The difference between the conventional KD and our $\mathcal{L}_{KL\_p}$ is that the latter also distills supervised class posterior from *extra self-supervision transformed images* from teacher to student. As shown in Table X, for online HSSAKD, three variants are (1) our core loss $\mathcal{L}^{S_1\sim S_2}_{aux\_SSAD} + \mathcal{L}^{S_1\sim S_2}_{KL\_q}$ for online learning and distilling self-supervision augmented distributions; (2) $\mathcal{L}^{S_1\sim S_2}_{aux\_SSAD} + \mathcal{L}^{S_1\sim S_2}_{KL\_q}$+DML [39]; (3) $\mathcal{L}^{S_1\sim S_2}_{aux\_SSAD} + \mathcal{L}^{S_1\sim S_2}_{KL\_q} + \mathcal{L}^{S_1\sim S_2}_{KL\_p}$. The difference between DML and our $\mathcal{L}^{S_1\sim S_2}_{KL\_p}$ is that the latter also performs mutual distillation of supervised class posterior from *extra self-supervision transformed images* between two online networks.

We can derive similar observations under these two scenarios. First, learning and distilling hierarchical self-supervision augmented knowledge through multiple auxiliary branches by loss $\mathcal{L}^S_{KL\_q}$ for offline HSSAKD and loss $\mathcal{L}^{S_1\sim S_2}_{aux\_SSAD} + \mathcal{L}^{S_1\sim S_2}_{KL\_q}$ for online HSSAKD substantially boosts the accuracy upon the original task loss $\mathcal{L}_{task}$. Moreover, we compare offline $\mathcal{L}^S_{KL\_p}$ with conventional KD, as well as online $\mathcal{L}^{S_1\sim S_2}_{KL\_p}$ with DML. The superior results over previous methods suggest that distilling probabilistic class posterior from those self-supervised transformed images is also beneficial to KD-based feature learning.

B. Effect of auxiliary branches

We propose to append several auxiliary branches to a network over various depths to learn and transfer diverse self-supervision augmented distributions extracted from hierarchical features. To examine the effectiveness of each auxiliary branch, we individually evaluate each one and drop others. As shown in Fig. 5, we can observe that each auxiliary branch is beneficial to performance improvements. Moreover, an auxiliary branch attached in the deeper layer often achieves a more accuracy gain than one in the shallower layer. This can be attributed to more informative semantic knowledge encoded in highly abstract features output from deep layers. Finally, we can aggregate all auxiliary branches to maximize accuracy gains, suggesting that it is meaningful to learn and transfer self-supervision augmented knowledge from *hierarchical feature maps*.

C. Effect of four different knowledge forms for offline HSSAKD

In Section IV-B1, we have examined the effectiveness of four different auxiliary tasks for training a single network. To further verify the efficacy, we employ four different distributions referred in Section III-C derived from these auxiliary tasks as knowledge to conduct offline KD. As shown Table XI, distilling our proposed self-supervision augmented distribution as auxiliary knowledge can significantly enhance performance and outperform other alternative distributions by large margins. Compared with others, it is demonstrated as richer knowledge that encodes more meaningful information derived from the joint task of supervised and self-supervised tasks.

D. Effect of various self-supervised auxiliary tasks

In this paper, we investigate three self-supervised classification tasks over the input image:

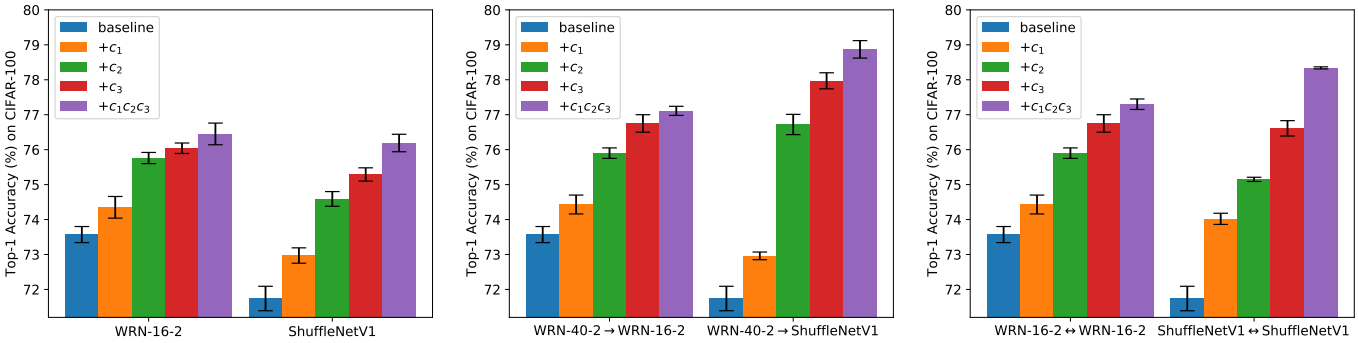
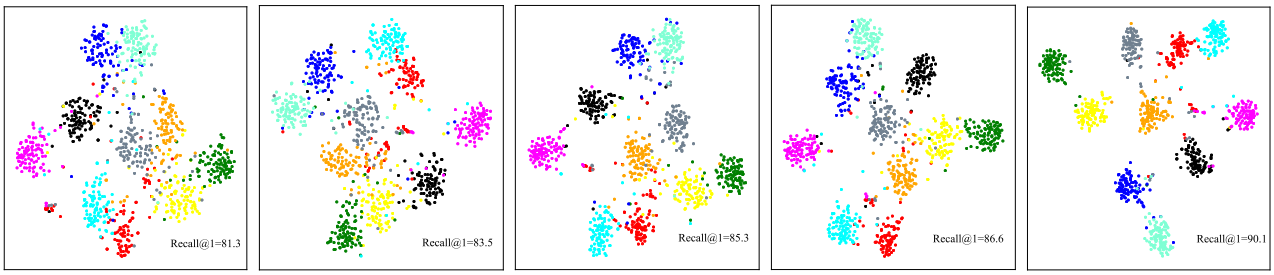


Fig. 5: Ablation study of the effectiveness of various auxiliary branches for learning and distilling self-supervision augmented knowledge on CIFAR-100 under: (1) ***left***: train a single network with auxiliary branches; (2) ***middle***: conventional offline HSSAKD; (3) ***right***: online mutual HSSAKD.



(a) Baseline (b) KD [12] (c) CRD [23] (d) SSKD [28] (e) HSSAKD (Ours)

Fig. 6: T-SNE [66] visualization of feature embeddings learned from various KD methods over randomly sampled 10 classes from CIFAR-100. Moreover, we use Recall@1 to quantify intra-class variations of embedding space. Recall@k denotes the percentage of test images that have at least one image from the same class in k nearest neighbours in the embedding space.

- *6-way color channal permutation*: {RGB, RBG, GRB, GBR, BRG, BGR}
- *4-way jigsaw puzzles*: Jigsaw splits the input image into $2 \times 2 = 4$ non-overlapping patches. We can rearrange these 4 input patches and lead to $4! = 24$ possible permutations. We re-organize these patches into the original image format and forces the network to classify the correct permutation. To shrink the class space, we select 4 permutations with maximum Hamming distance following [24].
- *4-way random rotations*: $\{0°, 90°, 180°, 270°\}$

In theory, different self-supervised tasks would lead to different qualities of representation learning. We examine these three self-supervised tasks to construct self-supervision augmented distributions for offline HSSAKD, respectively. As shown in Table XII, we find that 4-way random rotation is a more favourable self-supervised pretext task. The observation is consistent with the previous SSKD [28].

E. Effect of the temperature τ

The temperature τ is a common hyper-parameter in KD to smooth the produced probability distributions. In theory, a larger τ would result in a smoother distribution. To search the best τ, we vary it from 1 to 5, which is a reasonable range verified by previous KD works [12], [65]. As shown in Table XIII, we find that HSSAKD is robust to the temperature τ and $\tau = 3$ may be a more suitable choice.

TABLE XIII: Top-1 accuracy (%) comparison towards ablation study of temperature T on the student networks of WRN-16-2 and ShuffleNetV1 supervised through the pre-trained teacher network WRN-40-2 for *offline HSSAKD* on CIFAR-100.

Temperature T	$T=1$	$T=2$	$T=3$	$T=4$	$T=5$
WRN-16-2	78.20	78.35	**78.67**	78.52	78.54
ShuffleNetV1	78.20	79.65	**80.11**	80.03	80.01

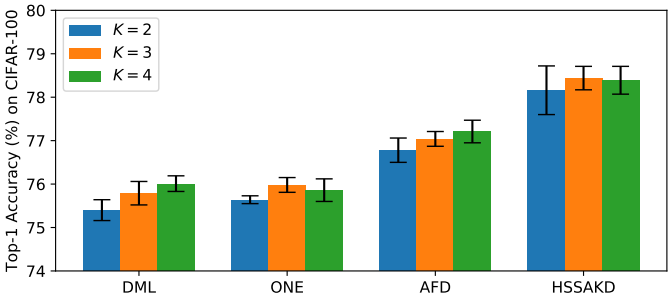


Fig. 7: Impact of the number of networks K for online KD.

F. Impact of the number of networks K for online KD

We investigate the performance of multi-student based online KD as the number of networks K increases. Fig. 7 plots the accuracy comparison among various online KD methods with the number of networks K changing from 2 to 4. We find that accuracy gains generally increase from $K = 2$ to

$K = 3$ but saturate at $K = 4$. Moreover, even in the case of the minimum training capacity of $K = 2$, our HSSAKD can still substantially beat other compared online KD methods with $K = 4$ networks. The results demonstrate the superiority of our self-supervision augmented distribution as an effective knowledge form.

G. Visualization of learned feature embeddings.

As shown in Fig. 6, we use t-SNE [66] to visualize pooled feature embeddings after the penultimate layer and report quantitative results on Recall@1. We can observe that our HSSAKD achieves higher visual separability and Recall@1 than other KD methods. The results demonstrate the efficacy of HSSAKD for generating a more discriminative feature space via distilling hierarchical self-supervised augmented knowledge. The better feature space further leads to a significant improvement of recognition accuracy.

H. Analysis of Training Overhead.

The offline KD conducts a two-stage pipeline that trains a teacher network and a student network sequentially. For training overhead, we compare our HSSAKD with the vanilla KD [12]. Compared with KD, extra training costs of our HSSAKD lies in attached auxiliary branches for learning and distillation. We take the widely used WRN-40-2-WRN-40-1 as a teacher-student pair for illustration. For the teacher, computation of the original WRN-40-2 is 330MFLOPs. After appending auxiliary branches, the computation increases to 770MFLOPs. For the student, computations of the original WRN-40-1 and its auxiliary branches augmented counterpart are 80MFLOPs and 190MFLOPs, respectively. In practice, the overall pipeline training time of our HSSAKD is about $2\times$ than the vanilla KD. However, HSSAKD outperforms the vanilla KD with a significant 3.1% accuracy gain, demonstrating the rationality of training costs.

VI. Conclusion

We propose a self-supervision augmented task for KD and further transfer such rich knowledge derived from hierarchical feature maps leveraging well-designed auxiliary branches. Our method achieves SOTA performance on the standard image classification benchmarks than other offline and online KD methods. It is also demonstrated that HSSAKD can guide the network to learn well-general feature representations for semantic recognition tasks due to the well-combined self-supervised auxiliary task. Due to the excellent performance, we believe that our HSSAKD may attract wide attention in the community of KD. In the future, we will apply our method to compress some networks for other visual recognition tasks, for example, object detection [67] and semantic segmentation [68].

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *TPAMI*, vol. 39, no. 6, pp. 1137–1149, 2016.

[3] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[4] C. Yang, Z. An, H. Zhu, X. Hu, K. Zhang, K. Xu, C. Li, and Y. Xu, "Gated convolutional networks with hybrid connectivity for image classification." in *AAAI*, 2020, pp. 12 581–12 588.

[5] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *CVPR*, 2018, pp. 6848–6856.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[7] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *ECCV*, 2018, pp. 116–131.

[8] Z. Ding, Y. Chen, N. Li, D. Zhao, Z. Sun, and C. P. Chen, "Bnas: Efficient neural architecture search using broad scalable architecture," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[9] C. Yang, Z. An, C. Li, B. Diao, and Y. Xu, "Multi-objective pruning for cnns using genetic algorithm," in *ICANN*, 2019, pp. 299–305.

[10] M. Lin, L. Cao, S. Li, Q. Ye, Y. Tian, J. Liu, Q. Tian, and R. Ji, "Filter sketch for network pruning," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[11] L. Cai, Z. An, C. Yang, Y. Yan, and Y. Xu, "Prior gradient mask guided pruning-aware fine-tuning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

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

[13] S. Ge, S. Zhao, C. Li, and J. Li, "Low-resolution face recognition in the wild via selective knowledge distillation," *TIP*, vol. 28, no. 4, pp. 2051–2062, 2018.

[14] S. Ge, Z. Luo, C. Zhang, Y. Hua, and D. Tao, "Distilling channels for efficient deep tracking," *TIP*, vol. 29, pp. 2610–2621, 2019.

[15] S. L. D. Z. Kangkai Zhang, Chunhui Zhang and S. Ge, "Student network learning via evolutionary knowledge distillation," *TCSVT*, vol. 32, no. 4, pp. 2251–2263, 2022.

[16] D. Yang, Y. Zhou, W. Shi, D. Wu, and W. Wang, "Rd-iod: Two-level residual-distillation-based triple-network for incremental object detection," *TOMM*, vol. 18, no. 1, pp. 1–23, 2022.

[17] D. Yang, Y. Zhou, A. Zhang, X. Sun, D. Wu, W. Wang, and Q. Ye, "Multi-view correlation distillation for incremental object detection," *Pattern Recognition*, p. 108863, 2022.

[18] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *ICLR*, 2015.

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

[20] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *AAAI*, vol. 33, 2019, pp. 3779–3787.

[21] S. Ahn, S. X. Hu, A. C. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *CVPR*, 2019, pp. 9163–9171.

[22] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, "Correlation congruence for knowledge distillation," in *ICCV*, 2019, pp. 5007–5016.

[23] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *ICLR*, 2020, pp. 499–515.

[24] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *ECCV*. Springer, 2016, pp. 69–84.

[25] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *ICLR*, 2018.

[26] H. Lee, S. J. Hwang, and J. Shin, "Self-supervised label augmentation via input transformations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5714–5724.

[27] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *AISTATS*. PMLR, 2015, pp. 562–570.

[28] G. Xu, Z. Liu, X. Li, and C. C. Loy, "Knowledge distillation meets self-supervision," in *ECCV*, 2020, pp. 588–604.

[29] C. Yang, Z. An, L. Cai, and Y. Xu, "Hierarchical self-supervised augmented knowledge distillation," in *IJCAI*, 2021, pp. 1217–1223.

[30] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, 2017, pp. 7130–7138.

[31] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *CVPR*, 2019, pp. 3967–3976.

[32] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *ICCV*, 2019, pp. 1365–1374.
[33] H.-J. Ye, S. Lu, and D.-C. Zhan, "Distilling cross-task knowledge via relationship matching," in *CVPR*, 2020, pp. 12 396–12 405.
[34] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
[35] Y. Wang, C. Xu, C. Xu, and D. Tao, "Adversarial learning of portable student networks," in *AAAI*, vol. 32, no. 1, 2018.
[36] N. Passalis, M. Tzelepi, and A. Tefas, "Probabilistic knowledge transfer for lightweight deep representation learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
[37] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *AAAI*, vol. 34, no. 04, 2020, pp. 5191–5198.
[38] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *CVPR*, 2020, pp. 2339–2348.
[39] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *CVPR*, 2018, pp. 4320–4328.
[40] G. Song and W. Chai, "Collaborative learning for deep neural networks," in *NeurIPS*, 2018, pp. 1832–1841.
[41] X. Zhu, S. Gong *et al.*, "Knowledge distillation by on-the-fly native ensemble," in *NeurIPS*, 2018, pp. 7517–7527.
[42] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, "Online knowledge distillation with diverse peers." in *AAAI*, 2020, pp. 3430–3437.
[43] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, and P. Luo, "Online knowledge distillation via collaborative learning," in *CVPR*, 2020, pp. 11 020–11 029.
[44] I. Chung, S. Park, J. Kim, and N. Kwak, "Feature-map-level online adversarial knowledge distillation," in *ICML*. PMLR, 2020, pp. 2006–2015.
[45] C. Yang, Z. An, and Y. Xu, "Multi-view contrastive learning for online knowledge distillation," in *ICASSP*, 2021, pp. 3750–3754.
[46] C. Yang, Z. An, L. Cai, and Y. Xu, "Mutual contrastive learning for visual representation learning," in *AAAI*, 2022.
[47] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *CVPR*, 2020, pp. 6707–6717.
[48] P. Bhat, E. Arani, and B. Zonooz, "Distill on the go: Online knowledge distillation in self-supervised learning," in *CVPR Workshops*, 2021, pp. 2678–2687.
[49] Y. Yao, C. Liu, D. Luo, Y. Zhou, and Q. Ye, "Video playback rate perception for self-supervised spatio-temporal representation learning," in *CVPR*, 2020, pp. 6548–6557.
[50] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, "Video cloze procedure for self-supervised spatio-temporal learning," in *AAAI*, vol. 34, no. 07, 2020, pp. 11 701–11 708.
[51] K. Zhang, P. Yao, R. Wu, C. Yang, D. Li, M. Du, K. Deng, R. Liu, and T. Zheng, "Learning positional priors for pretraining 2d pose estimators," in *ACM MM Workshops*, 2021, pp. 3–11.
[52] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.
[53] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.
[54] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.
[55] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019, pp. 5693–5703.
[56] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger, "Multi-scale dense networks for resource efficient image classification," *ICLR*, 2018.
[57] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *TPAMI*, vol. 41, no. 8, pp. 1979–1993, 2018.
[58] K. N. Zagoruyko S, "Wide residual networks," in *BMVC*, 2016.
[59] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.
[60] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.
[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
[62] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *AISTATS*, 2011, pp. 215–223.
[63] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.
[64] L. Zhang, Y. Shi, Z. Shi, K. Ma, and C. Bao, "Task-oriented feature distillation," *NeurIPS*, vol. 33, pp. 14 759–14 771, 2020.
[65] D. Chen, J.-P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen, "Cross-layer distillation with semantic calibration," in *AAAI*, vol. 35, no. 8, 2021, pp. 7028–7036.
[66] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
[67] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," *NIPS*, vol. 30, 2017.
[68] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, "Cross-image relational knowledge distillation for semantic segmentation," in *CVPR*, 2022, pp. 12 319–12 328.

Chuanguang Yang received the B.Eng. degree from Shandong Normal University, Jinan, China, in 2018. He is currently pursuing a Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, China. His research interests include knowledge distillation, visual representation learning and image classification.

Zhulin An received the B.Eng. and M.Eng. degrees in computer science from Hefei University of Technology, Hefei, China, in 2003 and 2006, respectively and the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2010. He is currently with the Institute of Computing Technology, Chinese Academy of Sciences, where he became a Senior Engineer in 2014. His current research interests include optimization of deep neural network and lifelong learning.

Linhang Cai received the B.Eng. degree from Sun Yat-sen University, Guangzhou, China, in 2019 and the M.Eng. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2022. His research interests include knowledge distillation, network pruning and image classification.

Yongjun Xu is a professor at Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS) in Beijing, China. He received his B.Eng. and Ph.D. degree in computer communication from Xi'an Institute of Posts & Telecoms (China) in 2001 and Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China in 2006, respectively. His current research interests include artificial intelligence systems, and big data processing.