Hierarchical Self-supervised Augmented Knowledge Distillation

Chuanguang Yang[1,2] , **Zhulin An**[1*] , **Linhang Cai**[1,2] and **Yongjun Xu**[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
{yangchuanguang, anzhulin, cailinhang19g, xyj}@ict.ac.cn

Abstract

Knowledge distillation often involves how to define and transfer knowledge from teacher to student effectively. Although recent self-supervised contrastive knowledge achieves the best performance, forcing the network to learn such knowledge may damage the representation learning of the original class recognition task. We therefore adopt an alternative self-supervised augmented task to guide the network to learn the joint distribution of the original recognition task and self-supervised auxiliary task. It is demonstrated as a richer knowledge to improve the representation power without losing the normal classification capability. Moreover, it is incomplete that previous methods only transfer the probabilistic knowledge between the final layers. We propose to append several auxiliary classifiers to hierarchical intermediate feature maps to generate diverse self-supervised knowledge and perform the one-to-one transfer to teach the student network thoroughly. Our method significantly surpasses the previous SOTA SSKD with an average improvement of 2.56% on CIFAR-100 and an improvement of 0.77% on ImageNet across widely used network pairs. Codes are available at https://github.com/winycg/HSAKD.
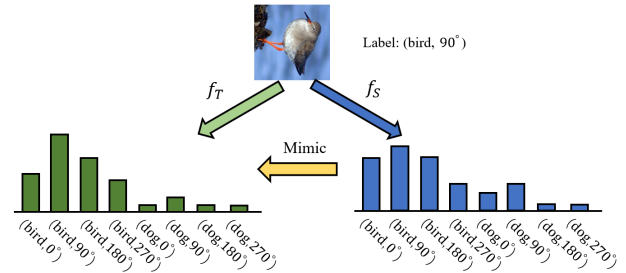
1 Introduction

Orthogonal to efficient network architecture designs [Yang *et al.*, 2019; Zhu *et al.*, 2019; Yang *et al.*, 2020], Knowledge Distillation (KD) [Hinton *et al.*, 2015] aims to transfer knowledge from a pre-trained high-capacity teacher network to a light-weight student network. The student's performance can often be improved significantly, benefiting from the additional guidance compared with the independent training. 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.

The original KD [Hinton *et al.*, 2015] minimizes the KL-divergence of predictive class probability distributions be-
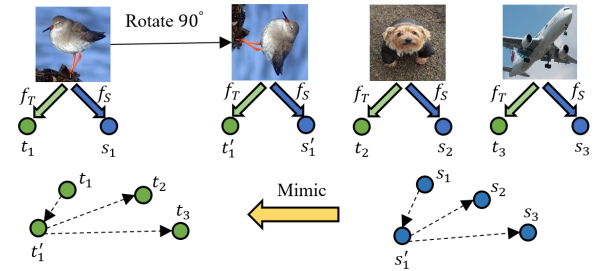
(a) Self-supervised contrastive relationship [Xu *et al.*, 2020].

(b) Our introduced self-supervised augmented distribution.

Figure 1: Difference of self-supervised knowledge between SSKD and our method. (a) SSKD applies contrastive learning by forcing the image and its transformed version closed against other negative images in the feature embedding space. It defines the contrastive relationship as knowledge. (b) Our method unifies the original task and self-supervised auxiliary task into a joint task and defines the self-supervised augmented distribution as knowledge.

tween student and teacher networks, which makes intuitive sense to force the student to mimic how a superior teacher generates the final predictions. However, such a highly abstract dark knowledge ignores much comprehensive information encoded in hidden layers. Later works naturally proposed to transfer feature maps [Romero *et al.*, 2015] or their refined information [Zagoruyko and Komodakis, 2017; Heo *et al.*, 2019; Ahn *et al.*, 2019] 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 an inductive bias of the final solution. Beyond knowledge alignment limited in the individual sample, more recent works [Peng *et al.*, 2019;

---

*Corresponding author

Dataset	Baseline	+DA (Rotation)	+SAL (Rotation)
CIFAR-100	78.01	$77.75_{(\downarrow-0.26)}$	$\mathbf{79.76}_{(\uparrow+1.75)}$
TinyImageNet	63.69	$62.66_{(\downarrow-1.03)}$	$\mathbf{65.81}_{(\uparrow+2.12)}$

Table 1: Top-1 accuracy on ResNet-18 using rotation as a data augmentation (DA) and a self-supervised augmented label (SAL).

Tian *et al.*, 2020] leverage cross-sample correlations or dependencies in high-layer feature embedding space. Inspired by the recent success of self-supervised visual representational learning [Chen *et al.*, 2020], SSKD [Xu *et al.*, 2020] introduces an auxiliary self-supervised task to extract richer knowledge. As shown in Fig. 1a, SSKD proposes transferring cross-sample self-supervised contrastive relationships, making it achieve superior performance in the field of KD.

However, forcing the network to learn invariant feature representations among transformed images using a self-supervised pretext task with random rotations from $0°, 90°, 180°, 270°$ utilized in SSKD may destroy the original visual semantics (*e.g.* 6 v.s. 9). It would increase the difficulty of representation learning for semantic recognition tasks. As validated in Table 1, applying random rotation as an additional data augmentation degrades the classification performance, especially on more challenging TinyImageNet.

To effectively learn knowledge from self-supervised representation learning without interfering with the original fully-supervised classification task, we use a unified task by combining the label spaces of the original task and self-supervised task into a joint label space, as shown in Fig. 1b. This task is partly inspired by the previous seminal self-supervised representation learning [Gidaris *et al.*, 2018; Lee *et al.*, 2020]. We further introduce these prior works to explore more powerful knowledge for distillation. To verify the effectiveness of the self-supervised augmented label, we also conduct initial exploratory experiments on standard image classification in Table 1. We find that the performance can be significantly improved by SAL, which can be attributed to learned better feature representations from an extra well-combined self-supervised task. The good performance further motivates us to define the self-supervised augmented distribution as a promising knowledge for KD.

Another valuable problem lies in how to transfer the probabilistic knowledge between teacher and student effectively. Vanilla KD aligns probability distributions only in the final layer but ignores comprehensive knowledge. Feature-based distillation methods provide one-to-one matching between the same convolutional stages of teacher and student. However, matched feature maps may have different semantic abstractions and result in a negative supervisory effect [Passalis *et al.*, 2020]. Compared with feature information, the probability distribution is indeed a more robust knowledge for KD, especially when existing a large architecture gap between teacher and student [Tian *et al.*, 2020].

However, it is difficult to explicitly derive comprehensive probability distributions from hidden layers over the original architecture. Therefore a natural idea is to append several auxiliary classifiers to the network at various hidden layers to generate multi-level probability distributions from hierar-chical feature maps. It allows us to perform comprehensive one-to-one matching in hidden layers in terms of probabilistic knowledge. Moreover, it is also noteworthy that the gap of abstraction level of any matched distributions would be easily reduced due to the delicately-designed auxiliary classifiers.

We guide all auxiliary classifiers attached to the original network to learn informative self-supervised augmented distributions. Furthermore, we perform Hierarchical Self-supervised Augmented Knowledge Distillation (HSAKD) between teacher and student towards all auxiliary classifiers in a one-to-one manner. By taking full advantage of richer self-supervised augmented knowledge, the student can be guided to learn better feature representations. Note that all auxiliary classifiers are only used to assist knowledge transfer and dropped during the inference period. The overall contributions are summarized as follows:

- We introduce a self-supervised augmented distribution that encapsulates the unified knowledge of the original classification task and auxiliary self-supervised task as the richer dark knowledge for the field of KD.
- We propose a one-to-one probabilistic knowledge distillation framework by leveraging the architectural auxiliary classifiers, facilitating comprehensive knowledge transfer and alleviating the mismatch problem of abstraction levels when existing a large architecture gap.
- HSAKD significantly refreshes the results achieved by previous SOTA SSKD on standard image classification benchmarks. It can also learn well-general feature representations for downstream semantic recognition tasks.

2 Related Work

Knowledge Distillation. The seminal KD [Hinton *et al.*, 2015] popularizes the pattern of knowledge transfer with a soft probability distribution. Later methods further explore feature-based information encapsulated in hidden layers for KD, such as intermediate feature maps [Romero *et al.*, 2015], attention maps [Zagoruyko and Komodakis, 2017], gram matrix [Yim *et al.*, 2017] and activation boundaries [Heo *et al.*, 2019]. More recent works explore cross-sample relationship using high-level feature embeddings with various definitions of edge weight [Park *et al.*, 2019; Peng *et al.*, 2019; Tung and Mori, 2019]. The latest SSKD [Xu *et al.*, 2020] extracts structured knowledge from the self-supervised auxiliary task. Beyond knowledge exploration, [Ahn *et al.*, 2019; Tian *et al.*, 2020] maximize the mutual information between matched features. [Yang *et al.*, 2021] further extend this idea into online KD in a contrastive-based manner. To bridge the gap between the teacher and student, [Passalis *et al.*, 2020] introduce teacher assistant models for smoother KD. However, extra teacher models increase the complexity of the training pipeline. Therefore, we choose to append several well-designed auxiliary classifiers to alleviate the knowledge gap and facilitate comprehensive knowledge transfer.

Self-supervised Representational Learning (SRL). The seminal SRL popularizes the pattern that guides the network to learn which transformation is applied to a transformed image for learning feature representations. Typical transformations can be rotations [Gidaris *et al.*, 2018], jig-
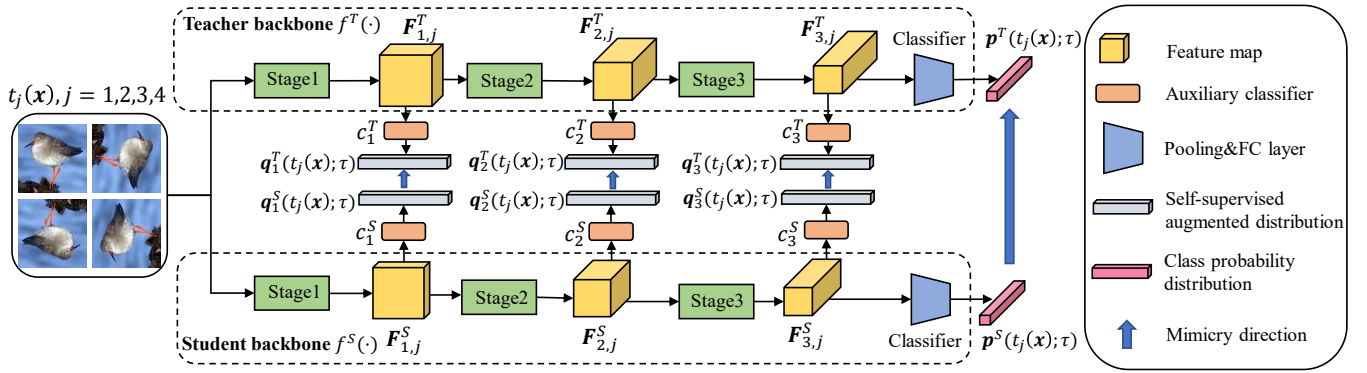
Figure 2: Overview of our proposed HSAKD. Both teacher and student networks are equipped with several auxiliary classifiers after various convolutional stages to capture diverse self-supervised augmented knowledge from hierarchical feature maps. Mimicry loss is applied from self-supervised augmented distributions of the student $\{q_l^S(t_j(\boldsymbol{x}); \tau)\}_{l=1}^L$ to corresponding that of the teacher $\{q_l^T(t_j(\boldsymbol{x}); \tau)\}_{l=1}^L$ generated from same feature hierarchies in a one-to-one manner. Following the conventional KD, we also force the mimicry from the class probability distribution of student $\boldsymbol{p}^S(t_j(\boldsymbol{x}); \tau)$ to that of the teacher $\boldsymbol{p}^T(t_j(\boldsymbol{x}); \tau)$. During the inference period, we only retain the student backbone $f^S(\cdot)$ and drop all auxiliary classifiers $\{c^S(\cdot)\}_{l=1}^L$. Therefore there has no extra inference cost compared with the original student network.

saw [Noroozi and Favaro, 2016] and colorization [Zhang *et al.*, 2016]. More recently, [Misra and Maaten, 2020; Chen *et al.*, 2020] learn invariant feature representations under self-supervised pretext tasks by maximizing the consistency of representations among various transformed versions of the same image. Both SSKD and our HSAKD are related to SRL. SSKD uses the latter SRL pattern to extract knowledge. In contrast, HSAKD combines the former classification-based pattern of SRL with the fully-supervised classification task to extract richer joint knowledge.

3 Method

3.1 Self-supervised Augmented Distribution

We present the difference between the original class probability distribution and self-supervised augmented distribution using a conventional classification network of CNN. A CNN can be decomposed into a feature extractor $\Phi(\cdot; \boldsymbol{\mu})$ and a linear classifier $g(\cdot; \boldsymbol{w})$, where $\boldsymbol{\mu}$ and $\boldsymbol{w}$ are weight tensors. Given an input sample $\boldsymbol{x} \in \mathcal{X}$, $\mathcal{X}$ is the training set, $\boldsymbol{z} = \Phi(\boldsymbol{x}; \boldsymbol{\mu}) \in \mathbb{R}^d$ is the extracted feature embedding vector, where d is the embedding size. We consider a conventional N-way object classification task with the label space $\mathcal{N} = \{1, \cdots, N\}$. The linear classifier attached with softmax normalization maps the feature embedding $\boldsymbol{z}$ to a predictive class probability distribution $\boldsymbol{p}(\boldsymbol{x}; \tau) = \sigma(g(\boldsymbol{z}; \boldsymbol{w})/\tau) \in \mathbb{R}^N$ over the label space, where σ is the softmax function, weight matrix $\boldsymbol{w} \in \mathbb{R}^{N \times d}$, τ is a temperature hyper-parameter to scale the smoothness of distribution.

We introduce an additional self-supervised task to augment the conventional supervised object class space. Learning such a joint distribution can force the network to generate more informative and meaningful predictions benefiting from the original and auxiliary self-supervised task simultaneously. Assuming that we define M various image transformations $\{t_j\}_{j=1}^M$ with the label space $\mathcal{M} = \{1, \cdots, M\}$, where t_1 denotes the identity transformation, *i.e.* $t_1(\boldsymbol{x}) = \boldsymbol{x}$. To effectively learn composite knowledge, we combine the

class space from the original supervised object recognition task and self-supervised task into a unified task. The label space of this task is $\mathcal{K} = \mathcal{N} \otimes \mathcal{M}$, here $\otimes$ is the Cartesian product. $|\mathcal{K}| = N * M$, where $|\cdot|$ is the cardinality of the label collection, $*$ denotes element-wise multiplication.

Given a transformed sample $\tilde{\boldsymbol{x}} \in \{t_j(\boldsymbol{x})\}_{j=1}^M$ by applying one transformation on $\boldsymbol{x}$, $\tilde{\boldsymbol{z}} = \Phi(\tilde{\boldsymbol{x}}; \boldsymbol{\mu}) \in \mathbb{R}^d$ is the extracted feature embedding vector, $\boldsymbol{q}(\tilde{\boldsymbol{x}}; \tau) = \sigma(g(\tilde{\boldsymbol{z}}; \boldsymbol{w})/\tau) \in \mathbb{R}^{N*M}$ is the predictive distribution over the joint label space $\mathcal{K}$, where weight tensor $\boldsymbol{w} \in \mathbb{R}^{(N*M) \times d}$. We use $\boldsymbol{p} \in \mathbb{R}^N$ to denote the normal class probability distribution and $\boldsymbol{q} \in \mathbb{R}^{N*M}$ to denote the self-supervised augmented distribution.

3.2 Auxiliary Architecture Design

It has been widely known that feature maps with various resolutions encode various patterns of representational information. Higher-resolution feature maps often present more fine-grained object details, while lower-resolution ones often contain richer global semantic information. To take full advantage of hierarchical feature maps encapsulated in a single network, we append several intermediate auxiliary classifiers into hidden layers to learn and distill hierarchical self-supervised augmented knowledge.

For ease of notation, we denote a conventional classification network as $f(\cdot)$, which maps a input sample $t_j(\boldsymbol{x}), j \in \mathcal{M}$ to the vanilla 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 $\mathcal{N}$. Modern CNNs typically utilize stage-wise convolutional blocks to gradually extract coarser features as the depth of the network increases. For example, popular ResNet-50 for ImageNet classification contains consecutive four stages, and extracted feature maps produced from various stages have different granularities and patterns. Assuming that a network contains L stages, we choose to append an auxiliary classifier after each stage, thus resulting in L classifiers $\{c_l(\cdot)\}_{l=1}^L$, where $c_l(\cdot)$ is the auxiliary classifier after l-th stage. $c_l(\cdot)$ is composed of stage-wise convolutional blocks, a global average pooling layer and a fully connected layer. Denoting the

extracted feature map after l-th stage as $\boldsymbol{F}_l$, we can obtain the self-supervised augmented distribution inferred by $c_l(\cdot)$ as $\boldsymbol{q}_l(t_j(\boldsymbol{x}); \tau) = \sigma(c_l(\boldsymbol{F}_l))/\tau) \in \mathbb{R}^{N*M}$ over the joint class space $\mathcal{K}$. The overall design of auxiliary classifiers over a 3-stage network is illustrated in Fig. 2 for example. ***The detailed design formulation of various auxiliary classifiers for a specific network can be found in our released codes***.

3.3   Training the Teacher Network

We denote the teacher backbone network as $f^T(\cdot)$ and L auxiliary classifiers as $\{c_l^T(\cdot)\}_{l=1}^L$. We conduct an end-to-end training process for preparing the teacher network. On the one hand, we train the $f^T(\cdot)$ with normal data $\boldsymbol{x}$ by the conventional Cross-Entropy (CE) loss to fit the ground-truth label $y \in \mathcal{N}$, and $\boldsymbol{p}^T(\boldsymbol{x}; \tau) = \sigma(f^T(\boldsymbol{x})/\tau) \in \mathbb{R}^N$ is the predictive class probability distribution. On the other hand, we aim to train L auxiliary classifiers $\{c_l^T(\cdot)\}_{l=1}^L$ for learning hierarchical self-supervised augmented distributions. Given an input sample $t_j(\boldsymbol{x})$, we feed the feature maps $\{\boldsymbol{F}_{l,j}^T\}_{l=1}^L$ generated from backbone $f^T(\cdot)$ to $\{c_l^T(\cdot)\}_{l=1}^L$, respectively. The predictive self-supervised augmented distribution inferred by the l-th classifier c_l^T is $\boldsymbol{q}_l^T(t_j(\boldsymbol{x}); \tau) = \sigma(c_l^T(\boldsymbol{F}_{l,j}^T))/\tau) \in \mathbb{R}^{N*M}$. We train all auxiliary classifiers using CE loss with self-supervised augmented labels across $\{t_j(\boldsymbol{x})\}_{j=1}^M$ as Eq. (1).

$$\mathcal{L}_{ce\_SAD}^T = \frac{1}{M} \sum_{j=1}^M \sum_{l=1}^L \mathcal{L}_{ce}(\boldsymbol{q}_l^T(t_j(\boldsymbol{x}); \tau), k_j) \qquad (1)$$

Where $\tau = 1$ and $\mathcal{L}_{ce}$ denotes the Cross-Entropy loss. For a bit abuse of notation, we use k_j to denote the self-supervised augmented label of $t_j(\boldsymbol{x})$ in joint class space $\mathcal{K}$. The overall loss for training a teacher is shown in Eq. (2).

$$\mathcal{L}_T = \mathbb{E}_{\boldsymbol{x} \in \mathcal{X}}[\mathcal{L}_{ce}(\boldsymbol{p}^T(\boldsymbol{x}; \tau), y) + \mathcal{L}_{ce\_SAD}^T] \qquad (2)$$

Note that the two losses in Eq. 2 have different roles. The first loss aims to simply fit the normal data for learning general classification capability. The second loss aims to generate additional self-supervised augmented knowledge by the existing hierarchical features derived from the backbone network. This method facilitates richer knowledge distillation benefiting from the self-supervised task beyond the conventional fully-supervised task.

3.4   Training the Student Network

We denote the student backbone network as $f^S(\cdot)$ and L auxiliary classifiers as $\{c_l^S(\cdot)\}_{l=1}^L$. We conduct an end-to-end training process under the supervision of the teacher network. The overall loss includes the task loss from ground-truth labels and the mimicry loss from the pre-trained teacher.

Task Loss
We force the $f^S(\cdot)$ to fit the normal data $\boldsymbol{x}$ as the task loss:

$$\mathcal{L}_{task} = \mathcal{L}_{ce}(\boldsymbol{p}^S(\boldsymbol{x}; \tau), y) \qquad (3)$$

Where $\boldsymbol{p}^S(\boldsymbol{x}; \tau) = \sigma(f^S(\boldsymbol{x})/\tau) \in \mathbb{R}^N$ is the predictive class probability distribution. We also had tried to force L auxiliary classifiers $\{c_l^S(\cdot)\}_{l=1}^L$ to learn the self-supervised augmented

distributions from the joint hard label of the original and self-supervised tasks by $\mathcal{L}_{ce\_SAD}^S$ as the additional loss:

$$\mathcal{L}_{ce\_SAD}^S = \frac{1}{M} \sum_{j=1}^M \sum_{l=1}^L \mathcal{L}_{ce}(\boldsymbol{q}_l^S(t_j(\boldsymbol{x}); \tau), k_j) \qquad (4)$$

Where $\boldsymbol{q}_l^S(t_j(\boldsymbol{x}); \tau) = \sigma(c_l^S(\boldsymbol{F}_{l,j}^S))/\tau) \in \mathbb{R}^{N*M}$, and $\boldsymbol{F}_{l,j}^S$ is the extracted feature map from the l-th stage of $f^S(\cdot)$ for the input $t_j(\boldsymbol{x})$. However, we empirically found that introducing loss (4) into the original task loss (3) damages the performance of student networks, as validated in Section 4.2.

Mimicry Loss
On the one hand, we consider transferring hierarchical self-supervised augmented distributions generated from L auxiliary classifiers of the teacher network to corresponding L auxiliary classifiers of the student network, respectively. The transfer performs in a one-to-one manner by KL-divergence loss D_{KL}. The loss is formulated as Eq. (5), where τ^2 is used to retain the gradient contributions unchanged [Hinton *et al.*, 2015].

$$\mathcal{L}_{kl\_q} = \frac{1}{M} \sum_{j=1}^M \sum_{l=1}^L \tau^2 D_{\mathrm{KL}}(\boldsymbol{q}_l^T(t_j(\boldsymbol{x}); \tau) \parallel \boldsymbol{q}_l^S(t_j(\boldsymbol{x}); \tau))$$
$$(5)$$

Benefiting from Eq. (5), one can expect that the student network gains comprehensive guidances by unified self-supervised knowledge and the original class full-supervised knowledge. The informative knowledge is derived from multi-scale intermediate feature maps encapsulated in hidden layers of the high-capacity teacher network.

On the other hand, we transfer the original class probability distributions generated from the final layer between teacher and student. Specifically, we transfer the knowledge derived from both the normal and transformed data $\{t_j(\boldsymbol{x})\}_{j=1}^M$, where $t_1(\boldsymbol{x}) = \boldsymbol{x}$. This loss is formulated as Eq. (6).

$$\mathcal{L}_{kl\_p} = \frac{1}{M} \sum_{j=1}^M \tau^2 D_{\mathrm{KL}}(\boldsymbol{p}^T(t_j(\boldsymbol{x}); \tau) \parallel \boldsymbol{p}^S(t_j(\boldsymbol{x}); \tau)) \quad (6)$$

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 mimicking the side product of predictive class probability distributions inferred from these transformed data from the teacher network is also beneficial for the self-supervised representational learning of student network, as validated in Section 4.2.

Overall loss.   We summarize the task loss and mimicry loss as the overall loss $\mathcal{L}_S$ for training the student network.

$$\mathcal{L}_S = \mathbb{E}_{\boldsymbol{x} \in \mathcal{X}}[\mathcal{L}_{task} + \mathcal{L}_{kl\_q} + \mathcal{L}_{kl\_p}] \qquad (7)$$

Following the wide practice, we set the hyper-parameter $\tau = 1$ in task loss and $\tau = 3$ in mimicry loss. Besides, we do not introduce other hyper-parameters.

4   Experiments

4.1   Experimental Settings

We conduct evaluations on standard CIFAR-100 and ImageNet [Deng *et al.*, 2009] benchmarks across the widely applied network families including ResNet [He *et al.*, 2016],

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
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	$75.23_{(\pm0.23)}$	$73.90_{(\pm0.44)}$	$70.91_{(\pm0.10)}$	$73.54_{(\pm0.26)}$	$75.21_{(\pm0.24)}$	$75.80_{(\pm0.46)}$	$75.83_{(\pm0.18)}$	$75.43_{(\pm0.33)}$
FitNet	$75.30_{(\pm0.42)}$	$74.30_{(\pm0.42)}$	$71.21_{(\pm0.16)}$	$75.37_{(\pm0.12)}$	$75.42_{(\pm0.34)}$	$75.41_{(\pm0.07)}$	$76.27_{(\pm0.18)}$	$76.91_{(\pm0.06)}$
AT	$75.64_{(\pm0.31)}$	$74.32_{(\pm0.23)}$	$71.35_{(\pm0.09)}$	$75.06_{(\pm0.19)}$	$74.08_{(\pm0.21)}$	$76.57_{(\pm0.20)}$	$76.51_{(\pm0.44)}$	$76.32_{(\pm0.12)}$
AB	$71.26_{(\pm1.32)}$	$74.55_{(\pm0.46)}$	$71.56_{(\pm0.19)}$	$74.31_{(\pm0.09)}$	$74.98_{(\pm0.44)}$	$75.87_{(\pm0.39)}$	$76.43_{(\pm0.09)}$	$76.40_{(\pm0.29)}$
VID	$75.31_{(\pm0.22)}$	$74.23_{(\pm0.28)}$	$71.35_{(\pm0.09)}$	$75.07_{(\pm0.35)}$	$75.67_{(\pm0.13)}$	$75.97_{(\pm0.08)}$	$76.24_{(\pm0.44)}$	$75.98_{(\pm0.41)}$
RKD	$75.33_{(\pm0.14)}$	$73.90_{(\pm0.26)}$	$71.67_{(\pm0.08)}$	$74.17_{(\pm0.22)}$	$75.54_{(\pm0.36)}$	$76.20_{(\pm0.06)}$	$75.74_{(\pm0.32)}$	$75.42_{(\pm0.25)}$
SP	$74.35_{(\pm0.59)}$	$72.91_{(\pm0.24)}$	$71.45_{(\pm0.38)}$	$75.44_{(\pm0.11)}$	$75.68_{(\pm0.35)}$	$76.35_{(\pm0.14)}$	$76.40_{(\pm0.37)}$	$76.43_{(\pm0.21)}$
CC	$75.30_{(\pm0.03)}$	$74.46_{(\pm0.05)}$	$71.44_{(\pm0.10)}$	$74.40_{(\pm0.24)}$	$75.66_{(\pm0.33)}$	$76.05_{(\pm0.25)}$	$75.63_{(\pm0.30)}$	$75.74_{(\pm0.18)}$
CRD	$75.81_{(\pm0.33)}$	$74.76_{(\pm0.25)}$	$\underline{71.83}_{(\pm0.42)}$	$75.77_{(\pm0.24)}$	$76.13_{(\pm0.16)}$	$76.89_{(\pm0.27)}$	$76.37_{(\pm0.23)}$	$76.51_{(\pm0.09)}$
SSKD	$\underline{76.16}_{(\pm0.17)}$	$\underline{75.84}_{(\pm0.04)}$	$70.80_{(\pm0.02)}$	$\underline{75.83}_{(\pm0.29)}$	$\underline{76.21}_{(\pm0.16)}$	$\underline{78.21}_{(\pm0.16)}$	$\underline{76.71}_{(\pm0.31)}$	$\underline{77.64}_{(\pm0.24)}$
Ours	$77.20_{(\pm0.17)}$	$77.00_{(\pm0.21)}$	$72.58_{(\pm0.33)}$	$77.26_{(\pm0.14)}$	$77.45_{(\pm0.21)}$	$78.79_{(\pm0.11)}$	$78.51_{(\pm0.20)}$	$79.93_{(\pm0.11)}$
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 2: Top-1 accuracy (%) comparison of SOTA distillation methods across various teacher-student pairs on CIFAR-100. All results are reproduced by ours using author-provided code. The numbers in **Bold** and <u>underline</u> 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.

WRN [Zagoruyko S, 2016], VGG [Simonyan and Zisserman, 2015], MobileNet [Sandler *et al.*, 2018] and ShuffleNet [Zhang *et al.*, 2018; Ma *et al.*, 2018]. Some representative KD methods including KD [Hinton *et al.*, 2015], FitNet [Romero *et al.*, 2015], AT [Zagoruyko and Komodakis, 2017], AB [Heo *et al.*, 2019], VID [Ahn *et al.*, 2019], RKD [Park *et al.*, 2019], SP [Tung and Mori, 2019], CC [Peng *et al.*, 2019], CRD [Tian *et al.*, 2020] and SOTA SSKD [Xu *et al.*, 2020] are compared. For a fair comparison, all comparative methods are combined with conventional KD by default, and we adopt rotations $\{0°, 90°, 180°, 270°\}$ as the self-supervised auxiliary task as same as SSKD. We use the standard training settings following [Xu *et al.*, 2020] and report the mean result with a standard deviation over 3 runs. ***The more detailed settings for reproducibility can be found in our released codes***.

4.2 Ablation Study

Effect of loss terms. As shown in Fig. 3 (left), applying hierarchical self-supervised augmented knowledge transfer through multiple auxiliary classifiers by loss $\mathcal{L}_{kl\_q}$ substantially boosts the accuracy upon the original task loss $\mathcal{L}_{task}$. We further compare $\mathcal{L}_{kl\_p}$ and $\mathcal{L}_{kd}$ upon $\mathcal{L}_{task} + \mathcal{L}_{kl\_q}$ to demonstrate the efficacy of transferring class probability distributions from additional transformed images. We find that $\mathcal{L}_{kl\_p}$ results in better accuracy gains than $\mathcal{L}_{kd}$, which suggests that transferring probabilistic class knowledge from those transformed images is also beneficial to feature representation learning. Finally, beyond the mimicry loss, we also explore whether the self-supervised augmented task loss $\mathcal{L}_{ce\_SAD}^S$ can be integrated into the overall task loss to train student networks. After adding $\mathcal{L}_{ce\_SAD}^S$ upon the above losses, the performance of the student network is dropped slightly. We speculate that mimicking soft self-supervised augmented distribution from the teacher by $\mathcal{L}_{kl\_q}$ is good
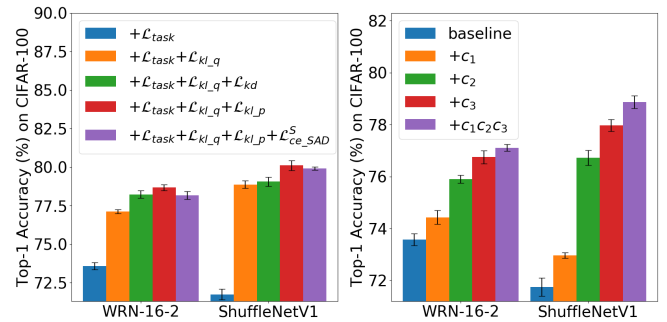

Figure 3: Ablation study of loss terms (***left***) and auxiliary classifiers (***right***) on the student networks WRN-16-2 and ShuffleNetV1 under the pre-trained teacher network WRN-40-2 on CIFAR-100.

enough to learn rich self-supervised knowledge. Extra learning from hard one-hot distribution by $\mathcal{L}_{ce\_SAD}^S$ may interfere with the process of self-supervised knowledge transfer.

Effect of auxiliary classifiers. We append several auxiliary classifiers to the network with various depths to learn and transfer diverse self-supervised augmented distributions extracted from hierarchical features. To examine this practice, we first individually evaluate each auxiliary classifier. As shown in Fig. 3 (right), we can observe that each auxiliary classifier is beneficial to performance improvements. Moreover, the auxiliary classifier attached in the deeper layer often achieves more accuracy gains than that in the shallower layer, which can be attributed to more informative semantic knowledge encoded in high-level features. Finally, using all auxiliary classifiers can maximize accuracy gains.

4.3 Comparison with State-Of-The-Arts

Results on CIFAR-100 and ImageNet. We compare our HSAKD with SOTA representative distillation methods

Teacher	Student	Acc	Teacher	Teacher*	Student	KD	AT	CC	SP	RKD	CRD	SSKD	Ours	Ours*
ResNet-34	ResNet-18	Top-1	73.31	75.48	69.75	70.66	70.70	69.96	70.62	71.34	71.38	_71.62_	72.16	**72.39**
		Top-5	91.42	92.67	89.07	89.88	90.00	89.17	89.80	90.37	90.49	_90.67_	90.85	**91.00**

Table 3: Top-1 accuracy (%) comparison on ImageNet. The compared results are from [Xu *et al.*, 2020].

Transferred Dataset	Baseline	KD	FitNet	AT	AB	VID	RKD	SP	CC	CRD	SSKD	Ours
CIFAR-100→ STL-10	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	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 4: Linear classification accuracy (%) of transfer learning on the student MobileNetV2 pre-trained using the teacher VGG-13.

Baseline	KD	CRD	SSKD	Ours
76.18	77.06	77.36	_77.60_	**78.45**

Table 5: Comparison of detection mAP (%) on Pascal VOC using ResNet-18 as the backbone pre-trained by various KD methods.

Percentage	KD	CRD	SSKD	Ours
25%	$65.15_{(\pm 0.23)}$	$65.80_{(\pm 0.61)}$	$\underline{67.82}_{(\pm 0.30)}$	$\mathbf{68.50}_{(\pm 0.24)}$
50%	$68.61_{(\pm 0.22)}$	$69.91_{(\pm 0.20)}$	$\underline{70.08}_{(\pm 0.13)}$	$\mathbf{72.18}_{(\pm 0.41)}$
75%	$70.34_{(\pm 0.09)}$	$\underline{70.98}_{(\pm 0.43)}$	$70.47_{(\pm 0.14)}$	$\mathbf{73.26}_{(\pm 0.11)}$

Table 6: 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.

across various teacher-student pairs with the same and different architectural styles on CIFAR-100 in Table 2 and on ImageNet in Table 3. Interestingly, using $\mathcal{L}_{ce\_SAD}^{T}$ as an auxiliary loss can improve the teacher accuracy of the final classification. Compared to the original 'Teacher', 'Teacher*' achieves an average gain of 4.09% across five teacher networks on CIFAR-100 and a top-1 gain of 2.17% for ResNet-34 on ImageNet. For downstream student networks, 'Teacher*' leads to an average improvement of 1.15% on CIFAR-100 and an improvement of 0.23% on ImageNet than 'Teacher'. These results indicate that our proposed loss $\mathcal{L}_T$ can improve the performance of a given network and produce a more suitable teacher for KD to learn a better student. Moreover, our HSAKD significantly outperforms the best-competing method SSKD across all network pairs with an average accuracy gain of 2.56% on CIFAR-100 and a top-1 gain of 0.77% on ImageNet. Compared with other SOTA methods, the superiority of HSAKD can be attributed to hierarchical self-supervised augmented knowledge distillation by the assistance with well-designed auxiliary classifiers.

Transferability of Learned Representations. Beyond the accuracy on the object dataset, we also expect 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 protocal [Tian *et al.*, 2020]. As shown in Table 4, we can observe that both SSKD and HSAKD achieve better accuracy than other comparative methods, demonstrating that using self-supervised auxiliary tasks for distillation is conducive to generating better feature representations. Moreover, HSAKD can significantly outperform the best-competing SSKD by 3.63% on STL-10 and 3.50% on TinyImageNet. The results verify that encoding the self-supervised auxiliary task as an augmented distribution in our HASKD has better supervision quality than the contrastive relationship in SSKD for learning good features.

Transferability 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. We use Faster-RCNN [Ren *et al.*, 2016] framework and follow the standard data preprocessing and finetuning strategy. The comparison on detection performance is shown in Table 5. Our method outperforms the original baseline by 2.27% mAP and the best-competing SSKD by 0.85% mAP. These results verify that our method can guide a network to learn better feature representations for semantic recognition tasks.

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 strategy for each few-shot setting, while maintaining the original test set. As shown in Table 6, our method can consistently surpass others 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 our method can effectively learn general feature representations from limited data. In contrast, previous methods often focus on mimicking the inductive bias from intermediate feature maps or cross-sample relationships, which may overfit the limited set and generalize worse to the test set.

5 Conclusion

We propose a self-supervised augmented task for KD and further transfer such rich knowledge derived from hierarchical feature maps leveraging well-designed auxiliary classifiers. Our method achieves SOTA performance on the standard image classification benchmarks in the field of KD. It can guide the network to learn well-general feature representations for semantic recognition tasks. Moreover, it has no hyper-parameters to be tuned and is easy to implement.

References

[Ahn *et al.*, 2019] Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *CVPR*, pages 9163–9171, 2019.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICML*, pages 1597–1607, 2020.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

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

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Heo *et al.*, 2019] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI*, volume 33, pages 3779–3787, 2019.

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

[Lee *et al.*, 2020] Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Self-supervised label augmentation via input transformations. In *ICML*, pages 5714–5724. PMLR, 2020.

[Ma *et al.*, 2018] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, pages 116–131, 2018.

[Misra and Maaten, 2020] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, pages 6707–6717, 2020.

[Noroozi and Favaro, 2016] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84, 2016.

[Park *et al.*, 2019] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019.

[Passalis *et al.*, 2020] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *CVPR*, pages 2339–2348, 2020.

[Peng *et al.*, 2019] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *ICCV*, pages 5007–5016, 2019.

[Ren *et al.*, 2016] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2016.

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

[Sandler *et al.*, 2018] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[Tian *et al.*, 2020] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *ICLR*, 2020.

[Tung and Mori, 2019] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, pages 1365–1374, 2019.

[Xu *et al.*, 2020] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *ECCV*, pages 588–604, 2020.

[Yang *et al.*, 2019] Chuanguang Yang, Zhulin An, Chao Li, Boyu Diao, and Yongjun Xu. Multi-objective pruning for cnns using genetic algorithm. In *ICANN*, pages 299–305, 2019.

[Yang *et al.*, 2020] Chuanguang Yang, Zhulin An, Hui Zhu, Xiaolong Hu, Kun Zhang, Kaiqiang Xu, Chao Li, and Yongjun Xu. Gated convolutional networks with hybrid connectivity for image classification. In *AAAI*, pages 12581–12588, 2020.

[Yang *et al.*, 2021] Chuanguang Yang, Zhulin An, and Yongjun Xu. Multi-view contrastive learning for online knowledge distillation. In *ICASSP*, pages 3750–3754, 2021.

[Yim *et al.*, 2017] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, pages 7130–7138, 2017.

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

[Zagoruyko S, 2016] Komodakis N Zagoruyko S. Wide residual networks. In *BMVC*, 2016.

[Zhang *et al.*, 2016] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666, 2016.

[Zhang *et al.*, 2018] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018.

[Zhu *et al.*, 2019] Hui Zhu, Zhulin An, Chuanguang Yang, Kaiqiang Xu, Erhu Zhao, and Yongjun Xu. Eena: efficient evolution of neural architecture. In *ICCV Workshops*, 2019.