Generalization Matters: Loss Minima Flattening via Parameter Hybridization for Efficient Online Knowledge Distillation

Tianli Zhang[1], Mengqi Xue[2], Jiangtao Zhang[1], Haofei Zhang[1], Yu Wang[1], Lechao Cheng[3], Jie Song[1,†], and Mingli Song[1]
[1]Zhejiang University, [2]Hangzhou City University, [3]Zhejiang Lab
{zhangtianli,zhjgtao,haofeizhang,yu.wang,sjie,brooksong}@zju.edu.cn,
mqxue@zucc.edu.cn, chenglc@zhejianglab.com

Abstract

Most existing online knowledge distillation (OKD) techniques typically require sophisticated modules to produce diverse knowledge for improving students' generalization ability. In this paper, we strive to fully utilize multi-model settings instead of well-designed modules to achieve a distillation effect with excellent generalization performance. Generally, model generalization can be reflected in the flatness of the loss landscape. Since averaging parameters of multiple models can find flatter minima, we are inspired to extend the process to the sampled convex combinations of multi-student models in OKD. Specifically, by linearly weighting students' parameters in each training batch, we construct a Hybrid-Weight Model (HWM) to represent the parameters surrounding involved students. The supervision loss of HWM can estimate the landscape's curvature of the whole region around students to measure the generalization explicitly. Hence we integrate HWM's loss into students' training and propose a novel OKD framework via parameter hybridization (OKDPH) to promote flatter minima and obtain robust solutions. Considering the redundancy of parameters could lead to the collapse of HWM, we further introduce a fusion operation to keep the high similarity of students. Compared to the state-of-the-art (SOTA) OKD methods and SOTA methods of seeking flat minima, our OKDPH achieves higher performance with fewer parameters, benefiting OKD with lightweight and robust characteristics. Our code is publicly available at https://github.com/tianlizhang/OKDPH.

1. Introduction

Deep learning achieves breakthrough progress in a variety of tasks by constructing a large capacity network pre-trained on massive data [6]. In order to apply high-parameterized models in the real-world scene with limited resources, the knowledge distillation (KD) technique [12] aims to obtain a compact and effective student model guided by a large-scale teacher model for model compression. Based on the developments of KD, Zhang *et al*. [44] propose the concept of online knowledge distillation (OKD) to view all networks as students and achieve mutual learning from scratch through peer teaching, liberating the distillation process from the dependency on pre-trained teachers. Existing OKD methods mainly encourage students to acquire diverse and rich knowledge, including aggregating predictions [10, 33, 37], combining features [19, 22, 28], working with peers [39, 47], learning from group leaders [2], and receiving guidance from online teachers [39].

Nevertheless, these strategies focus on designing sophisticated architectures to exploit heterogeneous knowledge to enhance students' generalization, but they lack explicit constraints on generalization. The concept of generalization to deep models is the ability to fit correctly on previously unseen data [25], which can be reflected by the flatness of the loss landscape [14, 18]. Flatness is the landscape's local curvature, which is costly to direct calculate by the Hessian. Considering the setting of multiple students in OKD, we utilize the theory of multi-model fusion in parameter space [9] to estimate the local curvature by the linear combination of students' parameters (we call it a *hybrid-weight model*, which is expressed as HWM). More specifically, HWM is a stochastic convex combination of parameters of multiple students on different data augmentations, which can sample multiple local points on the landscape. Intuitively, HWM's loss reflects the upper and lower bounds of the local region's loss and estimates the curvature of the landscape. Minimizing HWM's loss flattens the region and forms a landscape with smaller curvature.

Based on the above observation, we propose a concise and effective OKD framework, termed *online knowledge*

†Corresponding author

distillation with parameter hybridization (OKDPH), to promote flatter loss minima and achieve higher generalization. We devise a novel loss function for students' training that incorporates the standard Cross-Entropy (CE) loss and Kullback-Leibler (KL) divergence loss, but also a supervised learning loss from HWM. Specifically, HWM is constructed in each batch by linearly weighting multiple students' parameters. The classification error of HWM explicitly measures the flatness of the region around students on the loss landscape, reflecting their stability and generalization. The proposed loss equals imposing stronger constraints on the landscape, guiding students to converge in a more stable direction. For intuitive understanding, we visualize the loss landscape of students obtained by different methods in Fig. 1. Our students converge to one broader and flatter basin (thus superior generalization performance), while the students obtained by DML [44] converge to different sharp basins, degrading the robustness and performance.

Unfortunately, directly hybridizing students' parameters can easily lead to the collapse of HWM due to the high nonlinearity of deep neural networks [26, 31]. Therefore, we restrict the differences between students through intermittent fusion operations to ensure the high similarity of multi-model parameters and achieve effective construction of HWM. Concretely, at regular intervals, we hybridize the parameter of HWM with each student and, conversely, assign the hybrid parameter to the corresponding student. This process shortens the distance between students, shown as very close loss trajectories of our students in Fig. 1. However, it will not reduce diversity because students receive different types of data augmentation, and they can easily become diverse during training. Our method pulls students in the same direction, plays the role of strong regularization, and obtains one lightweight parameter that performs well in various scenarios. The solution derived from our method is expected to integrate the dark knowledge from multiple models while maintaining a compact architecture and can be competent for resource-constrained applications.

To sum up, our contributions are organized as follows:

- Inspired by the theory of multi-model fusion, we innovatively extend traditional weight averaging to an on-the-fly stochastic convex combination of students' parameters, called a hybrid-weight model (HWM). The supervision loss of HWM can estimate the curvature of the loss landscape around students and explicitly measure the generalization.

- We propose a brand-new extensible and powerful OKD framework via parameter hybridization (OKDPH) for loss minima flattening, which flexibly adapts to various network architectures without modifying peers' structures and extra modules. It is the first OKD work that manipulates parameters.

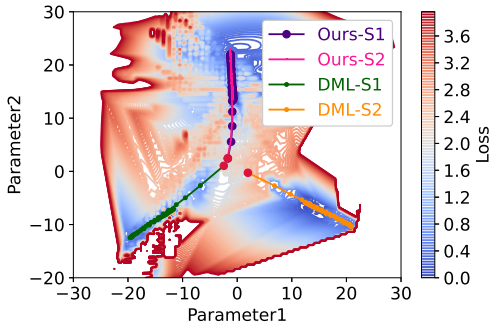

Figure 1. The loss landscape visualization of four students (Ours-S1 and Ours-S2 are obtained by our method, and DML obtains DML-S1 and DML-S2), which are ResNet32 [11] trained by the same settings on CIFAR-10 [20]. Four students start from the initial point (Red points in the center) and converge to three basins along different trajectories. The x-axis and y-axis represent the values of model parameters that PCA [23] obtains.

- Extensive experiments on various backbones demonstrate that our OKDPH can considerably improve the students' generalization and exceed the state-of-the-art (SOTA) OKD methods and SOTA approaches of seeking flat minima. Further loss landscape visualization and stability analysis verify that our solution locates in the region having uniformly low loss and is more robust to perturbations and limited data.

2. Related Work

Online Knowledge Distillation. Zhang *et al*. [44] propose deep mutual learning (DML) that enables students to share knowledge from each other's predictions to achieve distillation without teachers. KDCL [10], which is improved on DML, integrates the output of multiple students under different data augmentations as soft labels to guide students to optimize. In contrast to DML and KDCL, which promote collaboration between several students, ONE [46] employs a gating component to achieve distillation under the framework of one head and auxiliary branches. OKDDip [2] incorporates the attention mechanism into the multi-branch structure and guides students to learn from their auxiliary peers and the group leader. To constrain hidden representation between sub-networks, FFL [19] constructs a feature fusion module to improve the distillation effect, while PCL [39] builds a temporal mean model to act as an online teacher to produce more stable predictions. Ding *et al*. [5] design a knowledge refinery (KR) pipeline with decoupled labels to eliminate extra cumbersome teachers.

Generalization. Hinton *et al*. [13] apply the minimum description length principle to study the relationship between the generalization and the sharpness of minima.

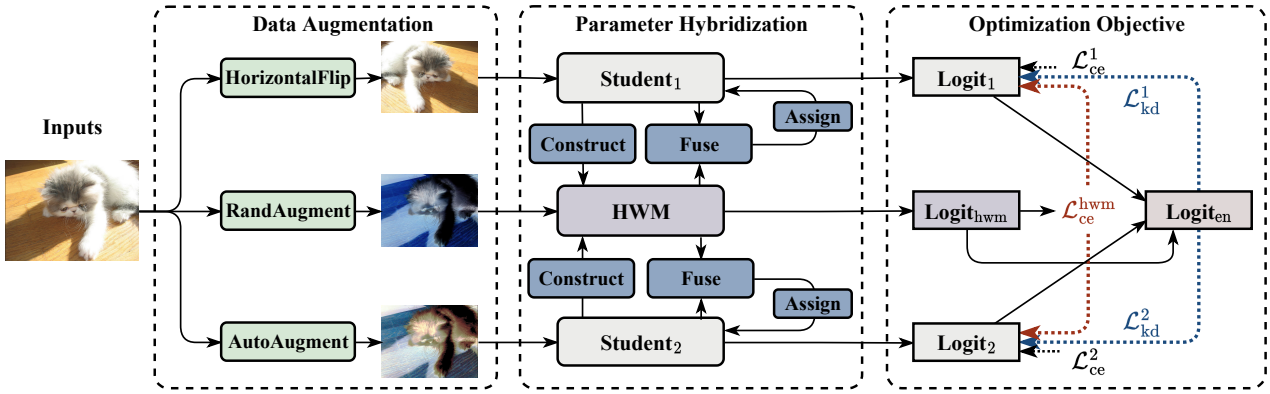

Figure 2. Framework of our OKDPH. Two students construct HWM by sampled convex parameter combinations in each training batch, and HWM's parameters are regularly fused with students. Two students and HWM's logits are obtained by feeding three types of data augmentations and are averaged to Logit$_{en}$. Each student's training loss consists of the classification loss $\mathcal{L}_{ce}$, $\mathcal{L}_{ce}^{hwm}$ and the KD loss $\mathcal{L}_{kd}$.

Hochreiter *et al*. [14] and Keskar *et al*. [18] propose that the flatness of the loss landscape basin nearby the solution is an indicator to measure the model generalization ability. According to Dziugaite *et al*. [7], the average empirical error is small if the model lies in a flat region of the parameter space. Langford *et al*. [21] construct the distribution of the model and improve its generalization by sensitivity analysis, that is, adding Gaussian noise with variance each time evaluating the data. Neyshabur *et al*. [25] associate sharpness with PAC-Bayes and believe network scales, such as norm and margin, also affect the generalization ability. In addition, Mobahi *et al*. [24] demonstrate how distillation can improve the generalization ability of networks through regularization and sparsity in Hilbert space.

Parameter Fusion. Frankle *et al*. [9] define a phenomenon called linear mode connectivity (LMC) as the parameters of two networks a path can connect with low loss. They hold that if two models can be linearly connected without barriers, they are inclined to be in the same basin and show more stability to noise. Model Soup [38] uses this principle to average parameters of multiple pre-trained models and achieves remarkable performance improvement on ImageNet. Neyshabur *et al*. [26] believe that two networks, even with the same random initialization, can be observed LMC barriers. Singh *et al*. [31] propose that since there is no one-to-one correspondence between well-trained network layers, achieving model fusion by direct average parameters is challenging. Tarvainen *et al*. [36] construct a mean teacher with more accurate labels by continuous students' exponential moving average (EMA), resulting in better test accuracy. SWA [16] achieves a wide and generalizable solution by weighted averaging the local minimum located in the border of areas with lower loss. Considering the relationship between the loss landscape's geometry and generalization, SAM [8] seeks parameters in neighborhoods with uniformly low loss.

3. Method

3.1. Vanilla Online Knowledge Distillation

Generally, vanilla OKD replaces the commonly used pre-trained teachers with peer student models. The training loss consists of the Cross-Entropy (CE) loss and the Knowledge Distillation (KD) loss. Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ be a training set containing N images and C categories of labels. The m-th student ($m \in \{1, \cdots, M\}$) obtains its output logits $\mathbf{z}^m \in \mathcal{R}^{\mathcal{C}}$ by feeding an augmentation of x_i. The classification loss is calculated by Cross-Entropy:

$$\mathcal{L}_{ce}\left(\mathbf{z}^m, y_i\right) = -\log \frac{\exp\left(\mathbf{z}_{y_i}^m\right)}{\sum_{c=1}^{C} \exp\left(\mathbf{z}_c^m\right)}, \tag{1}$$

where $y_i \in \{1, \cdots, C\}$ is the ground-truth label of the image x_i. The KD loss requires measuring the alignment of output distribution between models, which is usually achieved by KL divergence with temperature:

$$\mathcal{L}_{kd} = \tau^2 \mathcal{D}_{KL}\left(p^m, p^j\right) = \tau^2 \sum_{c=1}^{C} p_c^j \log \frac{p_c^j}{p_c^m}, \tag{2}$$

where $p^m, p^j \in \mathcal{R}^{\mathcal{C}}$ are the soft logits produced by a pair of students m and j. The soft logits are calculated by:

$$p^m = \sigma(\mathbf{z}^m/\tau) = \frac{\exp(\mathbf{z}^m/\tau)}{\sum_c \exp(\mathbf{z}_c^m/\tau)}, \tag{3}$$

where σ is the softmax function, and τ is the temperature.

3.2. Parameter Hybridization

It is known that a trained model is a point in weight space, and averaging multiple points leads to finding flatter

minima in the loss landscape. Considering the multi-model setting in OKD, we extend the average process to constructing a hybrid-weight model (HWM) using a sampled convex parameter combination of peer students. Formally, we create an HWM in each batch during training as follows:

$$\theta_{hwm}^t = \sum_{m=1}^{M} r_m^t \theta_m^t,\ \mathbf{r}^t = [r_1^t, ..., r_M^t] \sim \mathrm{Dir}\,(\alpha) \qquad (4)$$

where t represents the t-th training bath. θ_{hwm}^t and θ_m^t are HWM's and the m-th student's parameters at the t-th training batch, M is the number of students, and $\mathbf{r}^t$ is the weight vector that subjects to $\sum_{m=1}^{M} r_m^t = 1$.

$\mathrm{Dir}\,(\alpha)$ is the Dirichlet distribution parameterized by $\alpha \in \mathcal{R}^M$, which is commonly adopted as a prior distribution for multivariate sampling [1]. Dynamic sampling $\mathbf{r}^t$ subject to $\mathrm{Dir}\,(\alpha)$ can achieve the following effects: first, HWM can fully explore the parameter points of the region around students by each batch sampling; second, the concentration vector α can easily adjust the sampling probability of different points in this region. Consistent with the relevant works of averaging multiple models [9, 16, 38], we pay more attention to the parameter center of multiple models. We fix $\alpha = \mathbf{1} \in \mathcal{R}^M$ to gradually increase the sampling probability from the borderline models to the center of gravity. As shown in Fig. 3, the color of the triangles represents the probability distribution of the 3-dimensional Dirichlet distribution $\mathrm{Dir}\,([1, 1, 1])$. The darker the color, the higher the sampling probability.

Previous studies show that over-parameterized students, even under the supervision of KD loss, are prone to stay away from each other during training, resulting in the collapse of their parameter hybridization [26, 31]. So we require an additional operation to constrain the similarity between students to construct HWM effectively. Inspired by some works about parameter fusion [17, 36], we fuse HWM and students in a particular proportion at regular intervals during training:

$$\text{If mod}\,(t, \Delta) = 0:\ \theta_m^t = \gamma\theta_{hwm}^t + (1-\gamma)\theta_m^t, \qquad (5)$$

where Δ and γ are two hyper-parameters to represent the fusion interval and ratio, respectively. Δ can be set at the epoch or batch level, for instance, every epoch or every five batches. γ is generally taken as 0.5 or 1, which means the average of HWM and each student or directly replacing each student with HWM, respectively. When $\gamma = 1$, the role of the fusion is to achieve an instant parameter ensemble of students through HWM in the training process.

In short, constructing HWM and the fusion between HWM and students make up our parameter hybridization strategy. The former samples the points of the region around the students in the parameter space, and the latter controls the range of this region to prevent students from diverging.

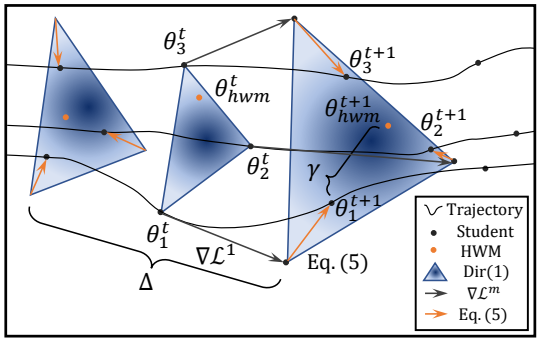

Figure 3. Schematic of the OKDPH parameter update.

3.3. Optimization Objective and Procedure

Fig. 2 shows the framework of our OKDPH in the case of two student models. Like some multi-model studies [10, 38], the student models and their HWM receive different data augmentation to produce informative and diverse predictions. Naturally, we synthesize knowledge in various scenarios by averaging the output logits of each model:

$$\mathbf{z}^{en} = \frac{1}{M+1}\left(\sum_{m=1}^{M} \mathbf{z}^m + \mathbf{z}^{hwm}\right), \qquad (6)$$

where $\mathbf{z}^m$ and $\mathbf{z}^{hwm}$ are the m-th students' and HWM's logits, respectively. The holistic knowledge of the ensemble logits $\mathbf{z}^{en}$ is further distilled into each student model by:

$$\mathcal{L}_{kd}(\mathbf{z}^m, \mathbf{z}^{en}) = \tau^2 \mathcal{D}_{KL}\left(\sigma\left(\frac{\mathbf{z}^m}{\tau}\right), \sigma\left(\frac{\mathbf{z}^{en}}{\tau}\right)\right), \qquad (7)$$

where $\mathcal{D}_{KL}$ is the KL divergence between the m-th student's logits $\mathbf{z}^m$ and the ensemble logits $\mathbf{z}^{en}$.

As mentioned in the previous section, the sampled HWM represents surrounding parameters in the space spanned by peer students. The HWM's classification loss $\mathcal{L}_{ce}^{hwm}$ for multiple consecutive training batches reflects upper and lower bounds on the loss in the region around students. Hence we integrate $\mathcal{L}_{ce}^{hwm}$ into each student's training to minimize the curvature of the region and flatten the loss landscape, enhancing students' generalization ability:

$$\mathcal{L}^m = \omega\mathcal{L}_{ce}^m + (1-\omega)\mathcal{L}_{ce}^{hwm} + \beta\mathcal{L}_{kd}(\mathbf{z}^m, \mathbf{z}^{en}), \qquad (8)$$

where $\mathcal{L}^m$ and $\mathcal{L}_{ce}^m$ are the m-th student's total loss and CE loss, respectively. The loss term ω can adjust the proportion of the current student's loss and its surrounding model's loss in the total classification loss. β represents the ratio of KD loss relative to CE loss, usually the same value as ω.

Our OKDPH procedure is summarized in Algorithm 1, and Fig. 3 shows the parameter update trajectories of three student models. Specifically, three students in the t-th training batch, *i.e.*, $\theta_1^t, \theta_2^t, \theta_3^t$, are optimized to $\theta_1^{t+1}, \theta_2^{t+1}$ and

Algorithm 1: OKD with Parameter Hybridization.

Input: Training data and labels: $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$; Number of training batches and students: T, M; Students and augmentations: $\{\theta_m^1\}_{m=1}^{M}$, $\{H_m\}_{m=1}^{M+1}$; Fusion interval and ratio: Δ, γ; Loss terms: ω, β.
Output: The best model parameter θ_{best}.

Initialize HWM θ_{hwm}^1 by Eq. (4);
foreach $t \in \{1, \cdots, T\}$ **do**
  Sample a batch of images and labels $X, Y \sim \mathcal{D}$;
  foreach $m \in \{1, \cdots, M\}$ **do**
    Predict logits output $\mathbf{z}^m = \theta_m^t(H_m(X))$;
  end
  Predict logits output $\mathbf{z}^{hwm} = \theta_{hwm}^t(H_{M+1}(X))$;
  Compute ensemble predictions $\mathbf{z}^{en}$ by Eq. (6);
  foreach $m \in \{1, \cdots, M\}$ **do**
    Optimize θ_m^t to θ_m^{t+1} by gradient of Eq. (8);
  end
  Construct HWM θ_{hwm}^{t+1} by Eq. (4);
  foreach $m \in \{1, \cdots, M\}$ **do**
    Update θ_m^{t+1} by Eq. (5);
  end
end
return Best model $\theta_{best} \in \{\theta_1^t, ..., \theta_M^t, \theta_{hwm}^t\}_{t=1}^{T}$.

θ_3^{t+1} by the gradient of Eq. (8). Then, their HWM θ_{hwm}^{t+1} is sampled on the triangle, formed by three student points. In the next training batch, the gradient of HWM's classification loss is passed to each student and guides the students to optimize to flatter loss minima in the back-propagation process. Notably, three students will gradually move away from each other in the parameter space, shown as a gradually larger triangle in Fig. 3. So every time the interval Δ is reached, we reduce the distance of three students by the fusion with HWM, shown as an orange arrow.

4. Experiment

We first describe the datasets and experimental settings in Sec. 4.1 and compare the proposed OKDPH with SOTA methods in Sec. 4.2. Then Sec. 4.3 measures the generalization by the loss landscape visualization and stability analysis. Also, we conduct the parameter sensitivity and ablation study in Sec. 4.4 and Sec. 4.5.

4.1. Experimental Settings

Datasets. **CIFAR-10** [20] and **CIFAR-100** [20], as two commonly-used small-scale datasets for OKD, are adopted to verify the effectiveness of OKDPH. Moreover, we also introduce a large-scale benchmark, *i.e.*, **ImageNet** [29], for validating our method with complex real-world images.

Backbones and Training Details. We use the framework of PyTorch [27] to implement our experiment in the setting of two student models and provide results with more students in the supplementary materials. For CIFAR-10 and CIFAR-100 datasets, we evaluate OKDPH on students with various backbones, including ResNet32 [11], ResNet110 [11], VGG16 [35], DenseNet40-12 [15], and WRN20-8 [43]. Each model receives multiple combinations of data augmentations, including random crop and normalization. Except for the above data augmentations, the two students and HWM also accept random horizontal flipping, Cutout [4], and Random Augment [3], respectively. The SGD optimizer [34] is adopted with a learning rate of 0.1 and a weight decay of $5e^{-4}$. The number of epochs and the batch size are set to 300 and 128, respectively. For the settings of ImageNet, we employ the standard ResNet18 [11] as the backbone and train 100 epochs with a learning rate of 0.1.

Baselines. The compared baselines include the mainstream methods of seeking flat minima and SOTA OKD methods, which aim at validating the superiority of our OKDPH in both two research fields. The former kind of methods includes exponential moving average (EMA [36]), stochastic weight averaging (SWA [16]), and sharpness aware minimization (SAM [8]). As for OKD methods, DML [44] and KDCL [10] encourage students' cooperation to promote mutual learning, while ONE [46], OKDDip [2], FFL [19], PCL [39], and KR [5] design additional modules or labels to produce and employ meaningful knowledge.

4.2. Results and Analysis

As shown in Tab. 1 and Tab. 2, we compare the top-1 accuracy of the proposed OKDPH with several SOTA methods on three datasets Like other OKD works and for fair comparisons, we report the best accuracy of one single model on the testing set and show the standard deviation by averaging five consecutive runs with a fixed random seed.

First, our method has the most outstanding performance on CIFAR-10, which is 0.64%, 1.00%, 0.20%, 0.50%, and 0.71% higher than the SOTA method on the backbones of ResNet32, ResNet110, VGG16, DenseNet40-12, and WRN20-8 respectively. Strikingly, the proposed method breaks through the 95% accuracy rate on ResNet32 in the OKD field for the first time. Secondly, on CIFAR-100, it can be observed that our OKDPH consistently beats all SOTA methods, which outperforms the sub-optimal PCL method by 1.67% and 1.16% on the ResNet110 and VGG16. It is no exaggeration to say that OKDPH raises the upper limit of OKD's performance on two CIFAR datasets. Also, our OKDPH performs significantly better than SOTA methods of seeking flat minima because our method utilizes extra multifarious knowledge brought by multiple models and the loss of distillation. Last but not least, we conduct experiments on ImageNet ILSVRC 2012 [29] to further verify the usefulness of OKDPH in scenarios involving sizable

Dataset	CIFAR-10					CIFAR-100				
Method	**ResNet32**	**ResNet110**	**VGG16**	**DenseNet40-12**	**WRN20-8**	**ResNet32**	**ResNet110**	**VGG16**	**DenseNet40-12**	**WRN20-8**
EMA	$93.71_{\pm 0.07}$	$94.78_{\pm 0.22}$	$94.07_{\pm 0.10}$	$93.25_{\pm 0.16}$	$94.68_{\pm 0.16}$	$71.29_{\pm 0.31}$	$75.26_{\pm 0.67}$	$72.64_{\pm 0.13}$	$70.82_{\pm 0.27}$	$76.58_{\pm 0.17}$
SWA	$94.00_{\pm 0.09}$	$94.79_{\pm 0.15}$	$94.56_{\pm 0.04}$	$93.18_{\pm 0.14}$	$94.65_{\pm 0.08}$	$72.06_{\pm 0.32}$	$75.26_{\pm 0.49}$	$74.69_{\pm 0.18}$	$71.10_{\pm 0.36}$	$76.31_{\pm 0.27}$
SAM	$94.41_{\pm 0.17}$	$94.91_{\pm 0.20}$	$95.13_{\pm 0.07}$	$93.17_{\pm 0.30}$	$95.13_{\pm 0.15}$	$72.54_{\pm 0.17}$	$75.50_{\pm 0.43}$	$74.55_{\pm 0.15}$	$70.74_{\pm 0.36}$	$75.86_{\pm 0.19}$
DML	$94.27_{\pm 0.08}$	$95.13_{\pm 0.14}$	$94.28_{\pm 0.13}$	$93.66_{\pm 0.05}$	$95.04_{\pm 0.14}$	$72.82_{\pm 0.17}$	$75.91_{\pm 0.23}$	$73.56_{\pm 0.05}$	$71.57_{\pm 0.21}$	$77.29_{\pm 0.13}$
ONE	$94.31_{\pm 0.07}$	$95.27_{\pm 0.13}$	$93.83_{\pm 0.07}$	$92.95_{\pm 0.06}$	$94.79_{\pm 0.03}$	$74.02_{\pm 0.29}$	$78.37_{\pm 0.30}$	$72.59_{\pm 0.15}$	$70.32_{\pm 0.23}$	$77.98_{\pm 0.33}$
KDCL	$93.91_{\pm 0.08}$	$95.11_{\pm 0.16}$	$94.24_{\pm 0.08}$	$93.87_{\pm 0.08}$	$95.27_{\pm 0.16}$	$71.83_{\pm 0.34}$	$78.28_{\pm 0.32}$	$73.98_{\pm 0.22}$	$71.35_{\pm 0.42}$	$77.97_{\pm 0.30}$
OKDDip	$94.19_{\pm 0.05}$	$95.16_{\pm 0.16}$	$93.72_{\pm 0.48}$	$93.03_{\pm 0.27}$	$94.61_{\pm 0.08}$	$71.71_{\pm 0.18}$	$77.60_{\pm 0.20}$	$72.71_{\pm 0.19}$	$70.30_{\pm 0.18}$	$77.75_{\pm 0.15}$
FFL	$94.32_{\pm 0.07}$	$95.33_{\pm 0.18}$	$93.92_{\pm 0.09}$	$92.93_{\pm 0.15}$	$92.16_{\pm 0.12}$	$73.39_{\pm 0.32}$	$77.61_{\pm 0.25}$	$72.95_{\pm 0.21}$	$70.78_{\pm 0.29}$	$69.53_{\pm 0.14}$
PCL	$94.20_{\pm 0.12}$	$94.85_{\pm 0.33}$	$94.22_{\pm 0.04}$	$92.25_{\pm 0.32}$	$92.57_{\pm 0.24}$	$72.86_{\pm 0.31}$	$78.33_{\pm 0.25}$	$73.54_{\pm 0.33}$	$69.95_{\pm 0.24}$	$77.88_{\pm 0.24}$
KR	$93.37_{\pm 0.19}$	$95.19_{\pm 0.26}$	$93.77_{\pm 0.14}$	$92.64_{\pm 0.13}$	$94.93_{\pm 0.18}$	$70.12_{\pm 0.07}$	$75.10_{\pm 0.61}$	$72.21_{\pm 0.16}$	$69.31_{\pm 0.43}$	$74.66_{\pm 0.16}$
OKDPH	$\mathbf{95.01_{\pm 0.09}}$	$\mathbf{96.28_{\pm 0.09}}$	$\mathbf{95.32_{\pm 0.05}}$	$\mathbf{94.34_{\pm 0.13}}$	$\mathbf{95.95_{\pm 0.09}}$	$\mathbf{74.10_{\pm 0.22}}$	$\mathbf{79.68_{\pm 0.29}}$	$\mathbf{75.56_{\pm 0.12}}$	$\mathbf{72.30_{\pm 0.26}}$	$\mathbf{78.88_{\pm 0.08}}$

Table 1. Top 1 accuracy (%) and standard deviation comparison of SOTA methods on CIFAR-10 and CIFAR-100.

Backbone	KD	DML	ONE	KDCL	OKDDip	FFL	PCL	Ours
ResNet18	69.51	69.82	70.18	69.60	69.93	68.85	70.22	**70.66**

Table 2. Accuracy (%) of several OKD methods on ImageNet.

datasets. Tab. 2 illustrates that OKDPH achieves the best top-1 accuracy compared with other OKD methods.

From the perspective of knowledge distillation, we believe that part of the reason for OKDPH's success is the broad scope of knowledge interaction. The carrier of knowledge transfer in OKD is usually the logits [10, 40, 44] and layers' representation [19, 41, 42]. We realize the former by Eq. (7) and extend the latter to the parameter level, which improves the output of all layers, not just the top-level layer. On the one hand, we conjecture that the parameter hybridization strategy can automatically integrate various dark knowledge [12, 30] encoded in multiple students, unlike other OKD baselines that rely on well-designed modules to generate effective knowledge [5, 32, 39, 46]. On the other hand, considering that students receiving different data augmentations can hybridize parameters without harming the accuracy, our OKDPH can be viewed as a regularization operation that regulates each layer's functionality and latent representations [24, 45], leading to one lightweight parameter that performs well in various scenarios. In summary, our method achieves higher performance with fewer parameters than the SOTA baselines in both fields, with a single model's test time and convenience.

4.3. Generalization Measurement

In this section, we conduct the generalization evaluation experiments using two traditional generalization measurements: the flatness of loss minima [14, 18] and the difference between the perturbed and empirical loss [21, 25].

4.3.1 Loss Landscape Visualization

We visualize multiple students obtained by various methods onto a single diagram to intuitively compare their differences in the flatness of the loss landscape. Specifically, we construct a set of high-dimensional vectors obtained by flattening students' parameters and use the PCA dimension reduction algorithm [23] to generate the two-dimensional coordinates. As shown in Fig. 4, the methods are Base (two students are independently trained only by the CE loss), DML [44], and KDCL [10] from left to right, and the color bar represents the training loss. Here we only display the above three methods because other OKD baselines introduce extra modules or branches, resulting in the vast difference in parameter quantities and thus failing the landscape comparison by PCA.

As each sub-diagram in Fig. 4 shows, our students converge to a broader and flatter basin (thus superior generalization performance). In contrast, other students converge to two sharp basins, resulting in a difference in generalization and impairing the overall stability of performance. Although students with different parameters are preferable for ensembles, which require diversified predictions to improve robustness, our approach differs from these ensemble methods. Assuming that our students fall into different basins, direct parameter hybridization can easily break down due to the high nonlinearity of deep neural networks. Therefore, our two students' loss trajectories are very close in the landscape, caused by the regular fusion operation (Eq. (5)) shortening students' distance. Considering our work aims to get one parameter that performs well in various scenarios, our students receiving different augmentations can become diverse quickly during training. Our method pulls them in the same direction, acting as strong regularization and improving the generalization. Overall, our OKDPH flattens loss minima and achieves the expected effect.

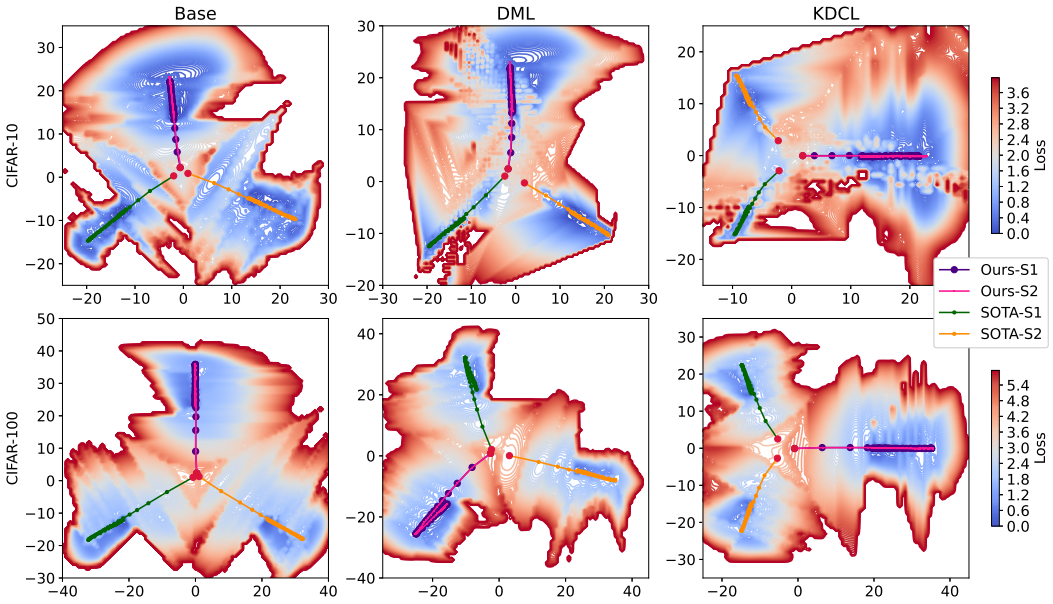

Figure 4. The loss landscape visualization of three methods (Base, DML [44], and KDCL [10] from left to right) compared with our method on two datasets (CIFAR-10 and CIFAR-100 [20] from top to bottom). Ours-S1 and Ours-S2 are the two students obtained by our method, and SOTA-S1 and SOTA-S2 are the two students obtained by other methods, both of which are ResNet32 [11] trained by the same settings. The x-axis and y-axis represent the values of model parameters by the PCA dimension reduction algorithm [23]. Each sub-diagram shows four students who start from the initial point (Red points in the center) and converge to three basins along different loss trajectories.

Dataset	Setting	ResNet32							VGG16						
		DML	ONE	KDCL	OKDDip	FFL	PCL	Ours	DML	ONE	KDCL	OKDDip	FFL	PCL	Ours
CIFAR-10	**Noisy**	83.02	83.44	82.95	83.70	83.38	83.18	**84.27**	83.96	83.09	83.91	83.34	83.00	83.07	**84.93**
	10%	81.48	81.64	81.15	81.50	81.52	81.13	**81.95**	80.56	76.64	81.49	78.17	79.26	79.44	**83.40**
	1%	38.66	41.08	41.20	38.68	38.55	33.17	**44.14**	41.03	35.93	42.79	35.75	40.28	39.73	**43.54**
CIFAR-100	**Noisy**	51.84	52.07	50.06	52.16	50.49	50.43	**52.89**	51.59	49.81	49.38	49.28	48.62	50.61	**53.63**
	10%	39.99	40.92	39.05	40.88	39.20	41.54	**41.83**	38.65	28.21	38.19	29.09	32.86	28.99	**42.59**
	1%	8.70	8.13	9.40	7.64	8.80	9.42	**9.71**	7.73	5.18	7.38	5.91	6.62	5.27	**8.32**

Table 3. Top 1 accuracy (%) comparison of several OKD methods in the context of noisy data (**Noisy**) and limited data (Sampling **10%** and **1%** of training data).

4.3.2 Stability Analysis

Except for the flatness of the loss landscape, stability analysis of models is also the primary tool for measuring generalization [21, 25]. In this subsection, we evaluate the performance of several OKD techniques in two contexts of noisy and limited data.

Specifically, the Gaussian random noise is added to the images in the training data, as shown below:

$$X = X + \text{Gussian}(\mu, \lambda), \tag{9}$$

where X is an image in the training data, and μ and λ are the mean and variance of the noise. Here we set μ and λ to 0 and 1 and calculate the accuracy on the same original test data. Besides, We randomly select 10% and 1% of the data from the training set for model training, and the test set remains unchanged. It is worth emphasizing that taking 1% of the training data on CIFAR-100 means that models can only see five images of the same kind, which is very challenging.

The results in Tab. 3 show that our method is far superior to other methods in well-known architectures. When 10% data are used to train VGG16, our OKDPH is much better than other OKD methods, 1.91% and 3.94% higher than the suboptimal method on CIFAR-10 and CIFAR-100, respectively. Compared with training on the complete dataset, our method shows more robust performance in the scenario of limited data, mainly due to the regularization effect brought

by our parameter hybridization. The overparameterization caused by various modules in other OKD methods will introduce more uncertainties, manifested as the phenomenon of overfitting, which reduces the generalization ability, thus leading to failure.

4.4. Parameter Sensitivity

As shown in Tab. 4, we explore the impact of four different hyperparameter $(\omega, \beta, \gamma, \Delta)$ values on the performance. We train two students with ResNet32 on CIFAR-10 for experiments and analyze the impact of one hyperparameter when the other three hyperparameters are fixed. The fusion proportion and interval γ, Δ directly constrain the distances between students since effective parameter hybridization requires a high similarity of multi-model parameters. The sub-tables in Tab. 4 show that the too-high or too-low similarity will lead to poor performance. When $\gamma = 0.0$, the performance is the worst, only 94.17%, which reflects the role of the fusion operation between the HWM and students.

ω	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
Acc	94.49	94.53	94.68	**95.01**	94.51

(a) Influence of ω with $\beta = 0.8, \gamma = 0.5, \Delta =$ 1e.

β	**0.0**	**0.2**	**0.5**	**0.8**	**1.0**
Acc	94.73	94.79	94.83	**95.01**	94.59

(b) Influence of β with $\omega = 0.8, \gamma = 0.5, \Delta =$ 1e.

γ	**0.0**	**0.2**	**0.5**	**0.8**	**1.0**
Acc	94.17	94.62	**95.01**	94.74	94.56

(c) Influence of γ with $\omega = 0.8, \beta = 0.8, \Delta =$ 1e.

Δ	**5b**	**200b**	**1e**	**2e**	**5e**
Acc	94.73	94.81	**95.01**	94.64	93.79

(d) Influence of Δ with $\omega = 0.8, \beta = 0.8, \gamma = 0.5$.

Table 4. Results (%) of OKDPH using ResNet32 with different values of four hyperparameters $(\omega, \beta, \gamma, \Delta)$ on CIFAR-10, where b and e are abbreviations for batch and epoch, respectively.

4.5. Ablation Study

In this section, we analyze the contribution of different components in our OKDPH to the final performance, including the KD loss $\mathcal{L}_{kd}$, the fusion of HWM and students (Eq. (5)), and the HWM's classification loss $\mathcal{L}_{ce}^{hwm}$. Tab. 5 shows the accuracy and performance improvement in four settings of the models trained under the backbone of ResNet32. As the basis of the whole distillation process, $\mathcal{L}_{kd}$ brings the greatest performance improvement.

Dataset	$\mathcal{L}_{ce}$	$\mathcal{L}_{kd}$	**Fuse**	$\mathcal{L}_{ce}^{hwm}$	**Acc (%)**
CIFAR-10	✓				93.26
	✓	✓			94.47 (+1.21)
	✓	✓	✓		94.75 (+0.28)
	✓	✓	✓	✓	95.01 (+0.26)
CIFAR-100	✓				72.76
	✓	✓			73.31 (+0.55)
	✓	✓	✓		73.60 (+0.29)
	✓	✓	✓	✓	74.10 (+0.50)

Table 5. Accuracy and performance improvement of four parts of our OKDPH using ResNet32 on CIFAR-10 and CIFAR-100.

It is necessary to fuse HWM's and students' parameters, which bring 0.28% and 0.29% improvements on CIFAR-10 and CIFAR-100, respectively. Due to the contribution of the $\mathcal{L}_{ce}^{hwm}$, our method achieves greater than 95% accuracy on CIFAR-10, breaking through the performance bottleneck and proving our idea's effectiveness.

Please refer to the supplementary materials for more experimental results, including experiments with training three or more students, extensive comparisons of generalization, and the display of hyperparameter values.

5. Conclusion

In this paper, we aim to explicitly measure the generalization in OKD and propose OKDPH to promote flatter loss minima and more stable convergence of students. A sampled hybrid-weight model, *i.e.*, HWM, is constructed in every training batch via the linear combination of all the students. Then, we adopt the supervision loss of HWM to guide students to converge to a flatter loss minima. Also, a novel fusion operation is designed to control the similarity of students to achieve effective parameter hybridization. Experiments with various backbones and datasets prove that OKDPH performs significantly better than SOTA methods of two fields. However, the limitation of OKDPH is that the parameter hybridization process can only be applied to homogeneous students. In the future, we strive to eliminate the limitation of multiple models and extend our method as a general optimizer to achieve broader applicability.

Acknowledgements. This work is funded by the National Key Research and Development Project (Grant No: 2022YFB2703100), National Natural Science Foundation of China (61976186, U20B2066, 62106235, 62106220), Ningbo Natural Science Foundation (2021J189), Fundamental Research Funds for the Central Universities (2021FZZX001-23), Open Research Projects of Zhejiang Lab (NO. 2019KD0AD01/018), and Exploratory Research Project of Zhejiang Lab(2022PG0AN01).

References

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. 4

[2] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3430–3437, 2020. 1, 2, 5

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 5

[4] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 5

[5] Qianggang Ding, Sifan Wu, Tao Dai, Hao Sun, Jiadong Guo, Zhang-Hua Fu, and Shutao Xia. Knowledge refinery: Learning from decoupled label. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7228–7235, 2021. 2, 5, 6

[6] Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021. 1

[7] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017. 3

[8] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. 3, 5

[9] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 1, 3, 4

[10] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020. 1, 2, 4, 5, 6, 7

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 5, 7

[12] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 1, 6

[13] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993. 2

[14] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997. 1, 3, 6

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[16] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 3, 4, 5

[17] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Amalgamating knowledge from heterogeneous graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15709–15718, 2021. 4

[18] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016. 1, 3, 6

[19] Jangho Kim, Minsung Hyun, Inseop Chung, and Nojun Kwak. Feature fusion for online mutual knowledge distillation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4619–4625. IEEE, 2021. 1, 2, 5, 6

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5, 7

[21] John Langford and Rich Caruana. (not) bounding the true error. *Advances in Neural Information Processing Systems*, 14, 2001. 3, 6, 7

[22] Chuanxiu Li, Guangli Li, Hongbin Zhang, and Donghong Ji. Embedded mutual learning: A novel online distillation method integrating diverse knowledge sources. *Applied Intelligence*, pages 1–14, 2022. 1

[23] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993. 2, 6, 7

[24] Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020. 3, 6

[25] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017. 1, 3, 6, 7

[26] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020. 2, 3, 4

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[28] Mary Phuong and Christoph H Lampert. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1355–1364, 2019. 1

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[30] Chengchao Shen, Mengqi Xue, Xinchao Wang, Jie Song, Li Sun, and Mingli Song. Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3504–3513, 2019. 6

[31] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020. 2, 3, 4

[32] Jie Song, Haofei Zhang, Xinchao Wang, Mengqi Xue, Ying Chen, Li Sun, Dacheng Tao, and Mingli Song. Tree-like decision distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13488–13497, 2021. 6

[33] Tongtong Su, Qiyu Liang, Jinsong Zhang, Zhaoyang Yu, Gang Wang, and Xiaoguang Liu. Attention-based feature interaction for efficient online knowledge distillation. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 579–588. IEEE, 2021. 1

[34] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 5

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[36] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 3, 4, 5

[37] Yankai Wang, Dawei Yang, Wei Zhang, Zhe Jiang, and Wenqiang Zhang. Adaptable ensemble distillation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1675–1679. IEEE, 2021. 1

[38] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. 3, 4

[39] Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10302–10310, 2021. 1, 2, 5, 6

[40] Mengqi Xue, Jie Song, Xinchao Wang, Ying Chen, Xingen Wang, and Mingli Song. Kdexplainer: A task-oriented attention model for explaining knowledge distillation. *arXiv preprint arXiv:2105.04181*, 2021. 6

[41] Xingyi Yang, Zhou Daquan, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. *arXiv preprint arXiv:2210.17409*, 2022. 6

[42] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 73–91. Springer, 2022. 6

[43] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[44] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018. 1, 2, 5, 6, 7

[45] Zhilu Zhang and Mert Sabuncu. Self-distillation as instance-specific label smoothing. *Advances in Neural Information Processing Systems*, 33:2184–2195, 2020. 6

[46] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31, 2018. 2, 5, 6

[47] Xuan Zhu, Wangshu Yao, and Kang Song. Online knowledge distillation with multi-architecture peers. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 1