Generalization Matters: Loss Minima Flattening via Parameter Hybridization for Efficient Online Knowledge Distillation
–*Supplementary Material*–

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

1. Implementation Details

In this section, we elaborate on the details of our experiment, including the dataset information and the complete training settings.

1.1. Datasets

To carry out the experiments, we adopt three datasets, CIFAR-10 [10], CIFAR100 [10], and ImageNet [15]. Information about the number of images and categories is shown below:

Task	Train	Test	Classes
CIFAR-10	50000	10000	10
CIFAR-100	50000	10000	100
ImageNet	1281167	50000	1000

Table 1. Detailed information of three involved datasets.

1.2. Training Settings

We adopt six different data augmentations: RandomCrop, RandomHorizontalFlip, RandomRotation, Cutout [6], RandomAugment [5], and AutoAugment [4]. As shown in Tab. 2, we compose them into five data transforms for training different models. The two student models and the HWM in our OKDPH are trained with three different transform compositions: 1, 3, 4 on CIFAR-10, 2, 3, 4 on CIFAR-100, and 1, 4, 5 on ImageNet. In addition, before data augmentation, all input images are resized to the same size of pixels, where 32×32 on CIFAR and 224×224 on ImageNet.

For CIFAR-10 and CIFAR-100, we evaluate OKDPH on students with various backbones, including ResNet32 [8], ResNet110 [8], VGG16 [17], DenseNet40-12 [9] and WRN20-8 [18]. According to the common settings in OKD, we use the SGD optimizer [16] with a learning rate of 0.1 decayed at the 150^{th} and 225^{th} epoch by a factor of 0.1 when updating the weights of students. The weight decay, the number of epochs, and batch sizes are set to $5e^{-4}$, 300, and 128, respectively.

For the settings of ImageNet, we employ the standard ResNet18 [8] as the backbone and train 100 epochs with a learning rate of 0.1, which is adjusted by the multi-step scheduler with the gamma of 0.1 and milestones of $\{30, 60, 90\}$ to obtain faster convergence. The weight decay of the SGD optimizer and batch sizes are set to $1e^{-4}$ and 256, respectively. It is worth emphasizing that the reported experimental results are the average accuracy of five consecutive runs with a fixed random seed of 42.

ID	Transform Composition		
1	RandomCrop	RandomHorizontalFlip	
2		RandomHorizontalFlip	RandomRotation
3		RandomHorizontalFlip	Cutout
4		RandomAugment	
5		AutoAugment	

Table 2. Details of five adopted transform compositions.

2. Results of Three or More Students

Tab. 3 shows the top 1 accuracy (%) comparison of several OKD methods in the case of multiple students or branches, where **#S/B** represents the number of students or branches. Specifically, DML, KDCL, and our OKDPH are multiple students, while the remaining methods are multiple branches. The three students and the HWM in our method are trained with the transform composition 1,3,4,5

†Corresponding author

Dataset	#S/B	ResNet32							VGG16						
		DML	ONE	KDCL	OKDDip	FFL	PCL	Ours	DML	ONE	KDCL	OKDDip	FFL	PCL	Ours
CIFAR-10	**2**	94.27	94.31	93.91	94.19	94.32	94.20	**95.01**	94.28	93.83	94.24	93.72	93.92	94.22	**95.32**
	3	94.31	94.36	94.06	94.22	94.49	94.22	**95.03**	94.33	93.92	94.26	93.84	93.96	94.74	**95.72**
	4	94.49	94.46	94.10	94.61	94.69	94.31	**95.13**	94.36	93.77	94.28	93.78	94.26	94.38	**95.78**
CIFAR-100	**2**	72.82	74.02	71.83	71.71	73.39	72.86	**74.10**	73.56	72.59	73.98	72.71	72.95	73.54	**75.56**
	3	73.24	74.16	71.71	74.27	74.09	73.09	**74.28**	73.73	72.72	73.79	73.04	72.61	75.89	**77.68**
	4	73.11	74.16	71.91	74.24	74.11	72.42	**74.56**	74.17	72.91	73.86	73.19	73.01	73.69	**78.06**

Table 3. Top 1 accuracy (%) comparison of several OKD methods in the case of multiple students or branches (**#S/B**).

Dataset	Setting	DML	ONE	KDCL	OKDDip	FFL	PCL	Ours
CIFAR-10	**Noisy**	81.37	80.62	81.53	80.85	80.94	80.57	**81.67**
	10%	82.07	80.19	81.70	80.49	79.83	80.31	**82.45**
	1%	45.21	45.02	44.47	44.24	42.96	44.89	**46.98**
CIFAR-100	**Noisy**	47.44	47.70	46.73	47.56	47.08	46.27	**48.62**
	10%	41.22	37.52	41.89	37.90	39.09	38.92	**42.24**
	1%	8.71	7.80	8.53	7.92	8.49	7.46	**8.74**

Table 4. Top 1 accuracy (%) comparison with the backbone of DenseNet40-12 in the context of noisy data (**Noisy**) and limited data (Sampling **10%** and **1%** of training data).

and 2,3,4,5 on CIFAR-10 and CIFAR-100, respectively, while in the case of four students, the fourth student is not used data augmentation.

The results in Tab. 3 show that the proposed method is improved steadily with more students and is far superior to other methods. Especially on CIFAR-100 using VGG16, our OKDPH is 2.14%, 2.36%, and 5.24% higher than the suboptimal method, respectively, in the case of 2, 3, and 4 students or branches.

3. Generalization Measurement

In this section, we qualitatively and quantitatively evaluate that our method obtains more generalized distillation effects, including visualization of the loss landscape, effects on noisy and limited data, and the generalization bound.

3.1. Loss Landscape Visualization

Fig. 2, Fig. 3, Fig. 4, and Fig. 5 show the loss landscape visualization of ResNet110, VGG16, DenseNet40-12, and WRN20-8, respectively. On all four backbones, our student models converge to a broader and flatter basin, while other methods converge to two basins with various flatness. Extensive loss landscape visualization on well-known architectures of the two datasets intuitively proves that the students we obtain have strong and consistent generalization abilities.

It is worth noticing that PCA can only be used on vectors of the same dimension. While other OKD methods, such as ONE [20] and OKDDip [3], are based on multi-branch network architectures, resulting in the vast difference in parameter quantities and thus failing in the landscape comparison by PCA. Consequently, we only compare OKDPH with DML and KDCL here.

3.2. Stability Analysis

As with the setting of the noisy and limited data in the main text, we additionally verify the proposed method's performance on the backbone of DenseNet40-12. As shown in Tab. 4, our method still outperforms the state-of-the-art methods.

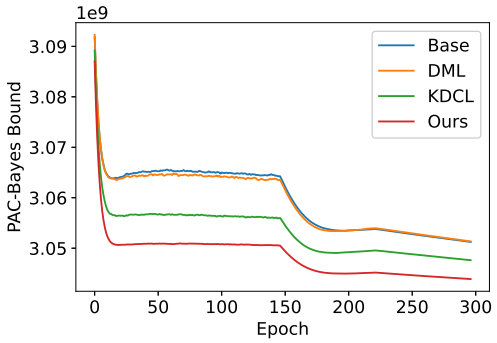

Figure 1. PAC-Bayes bound of models trained on VGG16 in different epochs.

Methods	CIFAR-10				CIFAR-100			
	Base	DML	KDCL	Ours	Base	DML	KDCL	Ours
ResNet32	97.78	97.57	97.69	**97.26**	103.03	102.43	102.66	**101.71**
ResNet110	239.55	239.82	239.34	**238.92**	249.44	248.63	248.17	**247.98**
VGG16	3051.24	3051.35	3047.62	**3043.88**	3089.95	3088.79	3078.59	**3068.47**
DenseNet40-12	38.57	38.34	38.80	**38.14**	45.19	43.65	41.77	**41.46**
WRN20-8	1376.46	1376.83	1376.24	**1374.61**	1391.57	1392.68	1384.58	**1383.76**

Table 5. PAC-Bayes bound ($\cdot 10^6$) of different backbones on CIFAR-10 and CIFAR-100.

3.3. PAC-Bayes Bound

In machine learning, the generalization bound [1, 14] is a probabilistic bound on the defect (generalization error), the maximum gap between the expected risk and the empirical risk. The PAC-Bayes theorem [2, 12, 13] bounds the expected error rate of a classifier chosen from a distribution Q in terms of the KL divergence from a prior fixed distribution P. Although PAC-Bayes bound in deep learning are often vacuous, they are the primary tool for measuring generalization.

Based on the above theory, We calculate the PAC-Bayes bound of various backbones on CIFAR-10 and CIFAR-100 to measure the generalization quantitatively. As shown in Tab. 5, our method achieves the lowest PAC-Bayes bound on all kinds of backbones, which shows that we get students with better generalization ability. In other words, our method obtains a model with a lower difference between the training error and the error outside the distribution.

Fig. 1 shows the PAC-Bayes bound of models trained on VGG16 in different epochs. With the continuous training of the model, the PAC-Bayes bound gradually decreases, which indicates that the model's generalization is improving. We notice that the models trained by the four methods have different degrees of overfitting. For example, the PAC-Bayes bound is increased slightly in the 175-th epoch. It can be seen that the overfitting amplitude of our method is significantly lower than that of the other three methods, which proves that parameter hybridization alleviates the overfitting phenomenon.

4. Hyperparameters

We describe the two formulas with four hyperparameters $(\omega, \beta, \gamma, \Delta)$ in the main text as follows:

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

when reach the interval Δ then do:

$$\theta_{m}^{t} = \gamma \theta_{hwm}^{t} + (1-\gamma)\theta_{m}^{t}, \quad (2)$$

where ω and β are the two loss terms. γ and Δ are the fusion proportion and interval. The fusion interval Δ can be set at the epoch or batch level, abbreviated as b and e, respectively. For example, 5b represents the fusion of student models and the HWM every five batches, and 1e represents the fusion of student models and the HWM every one epoch.

Tab. 6 shows the hyperparameter values of five different backbones on CIFAR-10 and CIFAR-100, where ResNet32-3 and ResNet32-4 represent the case of 3 and 4 students, respectively. Tab. 7 shows the hyperparameter values of experimental results in the context of noisy and limited data with the backbone of ResNet32, VGG16, and DenseNet40-12.

References

[1] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017. 3

[2] Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007. 3

[3] 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. 2

[4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 1

[5] 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. 1

[6] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1

[7] 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. 4, 5, 6

Dataset	HP	ResNet32	ResNet110	VGG16	DenseNet40-12	WRN20-8	ResNet32-3	ResNet32-4	VGG16-3	VGG16-4
CIFAR-10	ω	0.8	0.8	0.3	0.8	0.8	0.8	0.8	0.8	0.3
	β	0.8	0.8	0.3	0.8	0.8	0.8	0.8	0.8	0.3
	γ	0.5	0.5	1.0	0.5	0.5	0.5	0.5	0.5	1.0
	Δ	1e	5b	1e	5b	1e	1e	5b	1e	1e
CIFAR-100	ω	1.0	0.8	0.3	1.0	0.5	1.0	1.0	0.3	0.3
	β	0.5	0.8	0.3	0.5	0.5	0.5	0.5	0.3	0.3
	γ	1.0	0.5	1.0	0.5	0.5	1.0	1.0	1.0	1.0
	Δ	5b	5b	1e	5b	5b	5b	5b	3e	2e

Table 6. Optimal hyperparameter (**HP**) values of OKDPH with different backbones on CIFAR-10 and CIFAR-100, where b and e are abbreviations for batch and epoch, respectively. 5b and 1e represent the fusion of student models and the HWM every five batches and every epoch, respectively.

Dataset	Setting	ResNet32				VGG16				DenseNet40-12			
		ω	β	γ	Δ	ω	β	γ	Δ	ω	β	γ	Δ
CIFAR-10	**Noisy**	0.8	0.8	0.5	5b	0.3	0.3	1.0	1e	0.8	0.8	0.5	5b
	10%	1.0	0.5	1.0	5b	0.3	0.3	1.0	1e	0.8	0.8	0.5	5b
	1%	1.0	0.5	0.5	5b	0.3	0.3	1.0	1e	0.8	0.8	0.5	5b
CIFAR-100	**Noisy**	1.0	0.0	0.5	5b	0.3	0.3	1.0	1e	1.0	0.5	0.5	5b
	10%	0.5	0.5	1.0	5b	0.3	0.3	1.0	1e	1.0	0.5	0.5	5b
	1%	1.0	0.5	0.5	5b	0.8	0.8	1.0	1e	0.8	0.8	0.5	5b

Table 7. Optimal hyperparameter values of OKDPH with three backbones in the context of noisy and limited data, where b and e are abbreviations for batch and epoch, respectively. 5b and 1e represent the fusion of student models and the HWM every five batches and every epoch, respectively.

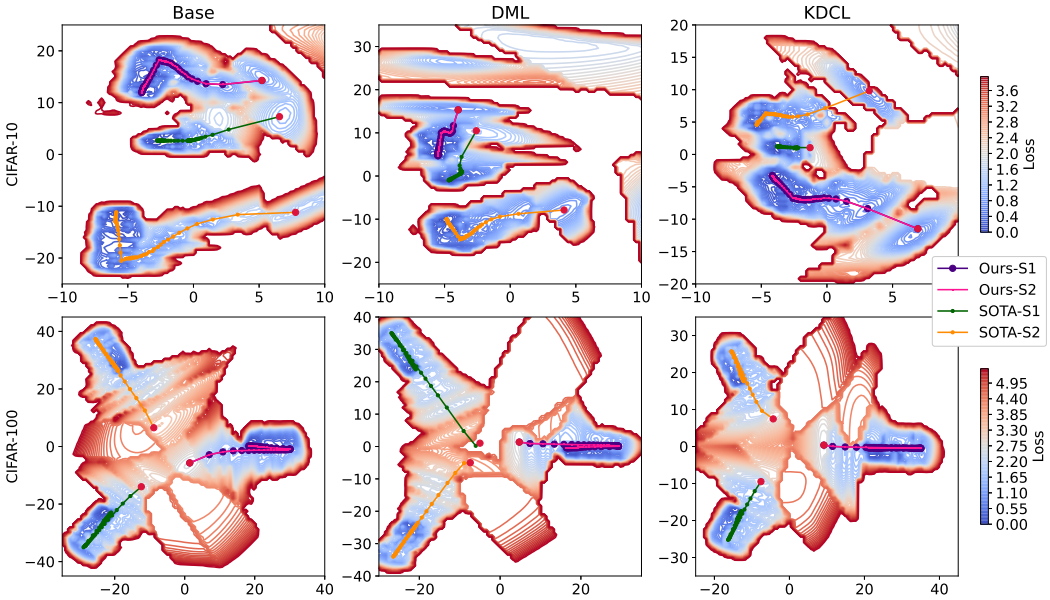

Figure 2. The loss landscape visualization of three methods (Base, DML [19], and KDCL [7] from left to right) compared with our method on two datasets (CIFAR-10 and CIFAR-100 [10] from top to bottom). Ours-S1 and Ours-S2 are the two students obtained by our method, and SOTA-S1 and SOTA-S2 are the two students obtained by other methods, both of which are ResNet32 [8] trained by the same settings. The x-axis and y-axis represent the values of model parameters by the PCA dimension reduction algorithm [11]. 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.

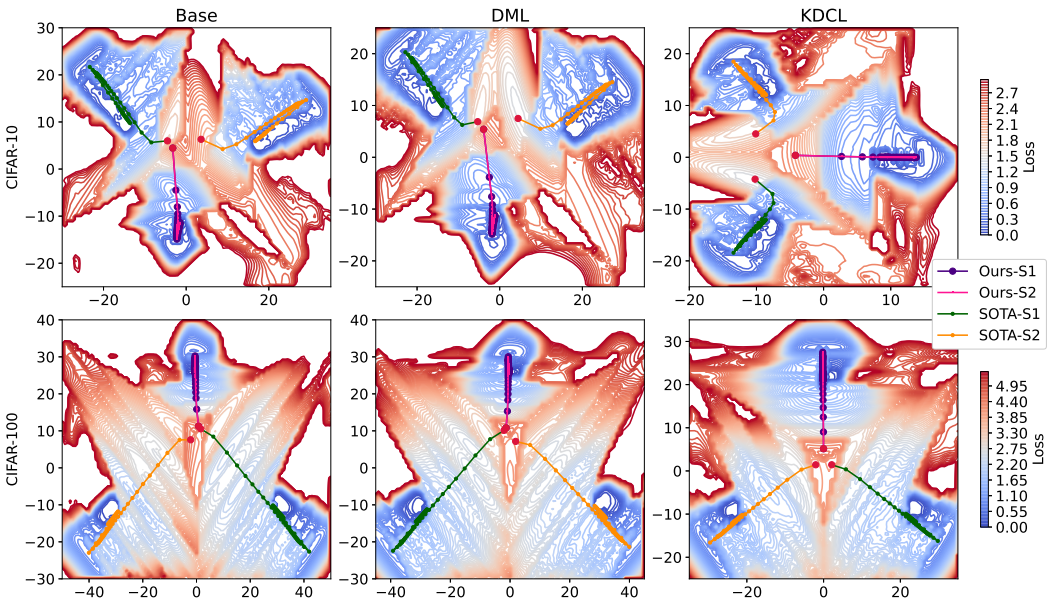

Figure 3. The loss landscape visualization of three methods (Base, DML [19], and KDCL [7] from left to right) compared with our method on two datasets (CIFAR-10 and CIFAR-100 [10] from top to bottom). Ours-S1 and Ours-S2 are the two students obtained by our method, and SOTA-S1 and SOTA-S2 are the two students obtained by other methods, both of which are VGG16 [17] trained by the same settings. The x-axis and y-axis represent the values of model parameters by the PCA dimension reduction algorithm [11]. 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.

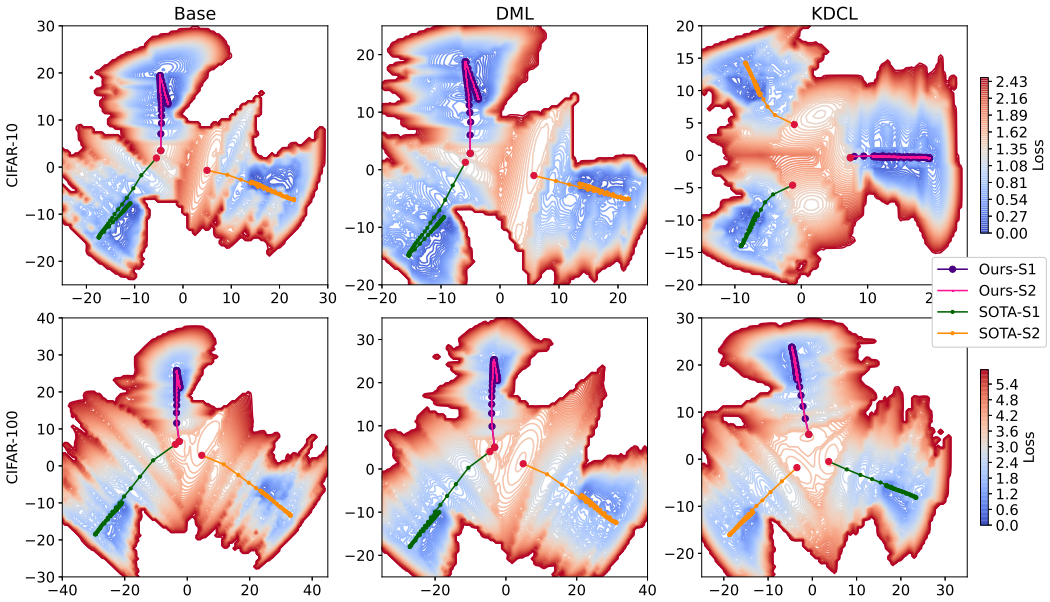

Figure 4. The loss landscape visualization of three methods (Base, DML [19], and KDCL [7] from left to right) compared with our method on two datasets (CIFAR-10 and CIFAR-100 [10] from top to bottom). Ours-S1 and Ours-S2 are the two students obtained by our method, and SOTA-S1 and SOTA-S2 are the two students obtained by other methods, both of which are DenseNet40-12 [9] trained by the same settings. The x-axis and y-axis represent the values of model parameters by the PCA dimension reduction algorithm [11]. 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.

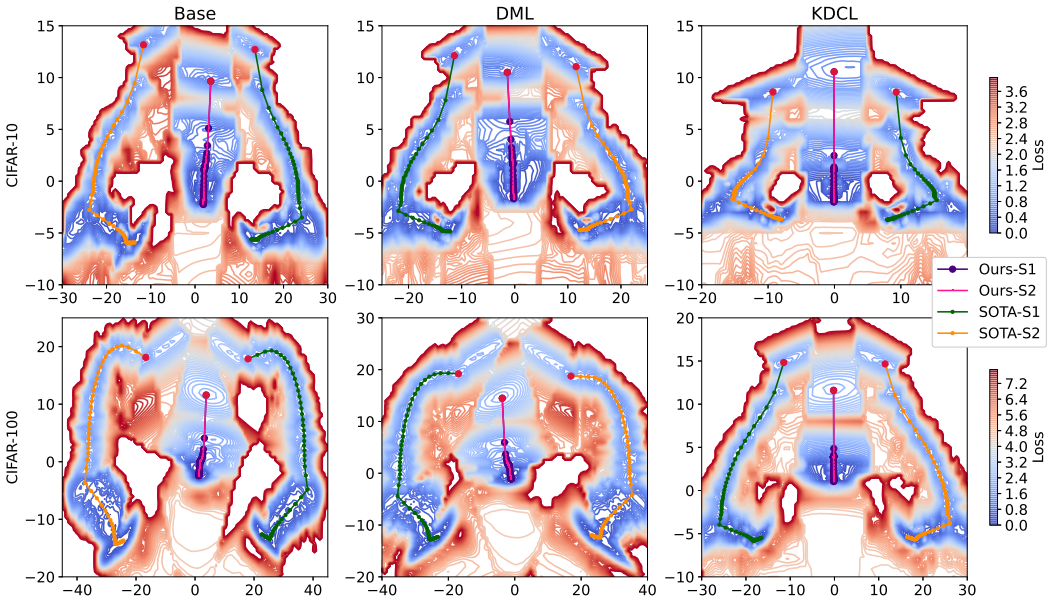

Figure 5. The loss landscape visualization of three methods (Base, DML [19], and KDCL [7] from left to right) compared with our method on two datasets (CIFAR-10 and CIFAR-100 [10] from top to bottom). Ours-S1 and Ours-S2 are the two students obtained by our method, and SOTA-S1 and SOTA-S2 are the two students obtained by other methods, both of which are WRN20-8 [18] trained by the same settings. The x-axis and y-axis represent the values of model parameters by the PCA dimension reduction algorithm [11]. 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.

[8] 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. 1, 4

[9] 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. 1, 5

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 4, 5, 6

[11] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993. 4, 5, 6

[12] David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998. 3

[13] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999. 3

[14] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401. PMLR, 2015. 3

[15] 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. 1

[16] 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. 1

[17] 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. 1, 5

[18] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 1, 6

[19] 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. 4, 5, 6

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