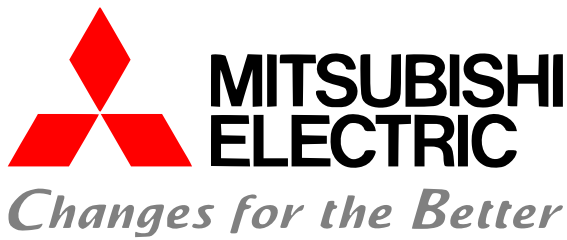
What Makes a "Good" Data Augmentation in Knowledge Distillation – A Statistical Perspective

Huan Wang[1,2,†] Suhas Lohit[2,*] Mike Jones[2] Yun Fu[1]

[1]Northeastern University, Boston, MA [2]MERL, Cambridge, MA

Project: https://huanwang-tech/Good-DA-in-KD

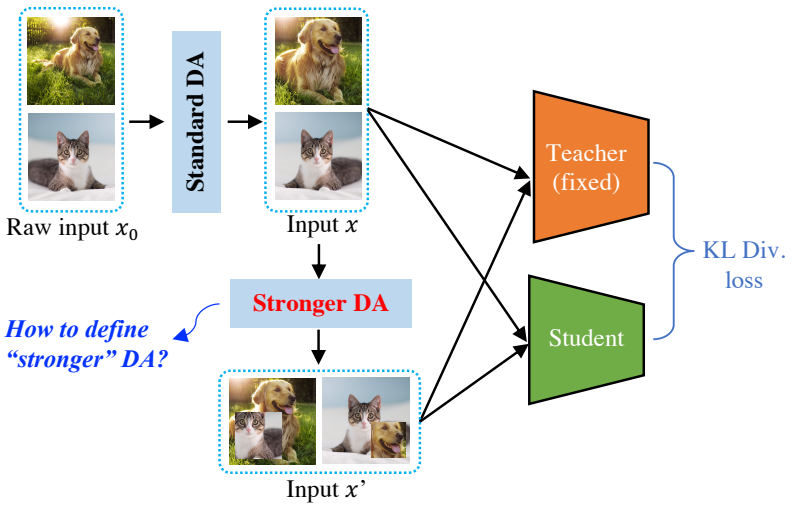
36th Conference on Neural Information Processing Systems (NeurIPS), 2022
New Orleans, Nov. 27-Dec.03, 2022



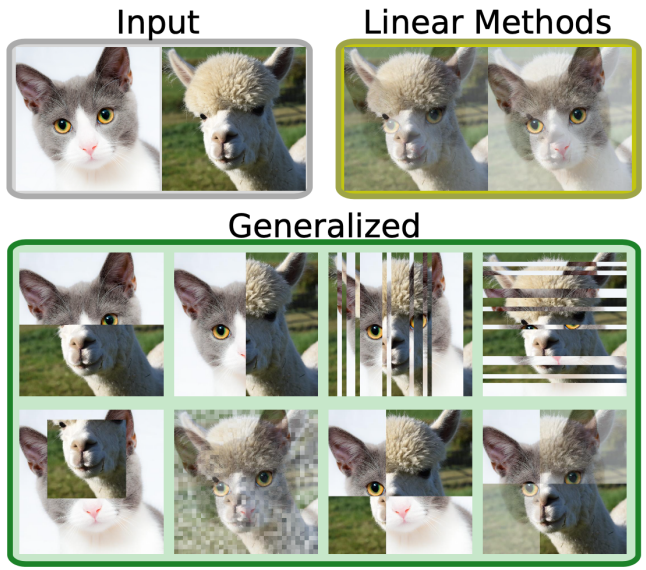
Background & Motivation

Background & Motivation: KD and DA

- **Knowledge distillation** (KD) (Hinton et al., 2014) is a generic neural network training framework that employs a (typically large and pretrained) *teacher* model to guide the learning of the (typically small) *student* model, by using the teacher's output as soft target for the student.
- **Data augmentation** (DA) (Shorten & Khoshgoftaar, 2019) is a large group of *input transformation* methods in machine learning, which can inflate the training sample size to curb overfitting.



(a) DA applied to KD in our paper



(b) DA samples (Summers & Dinneen, 2019)

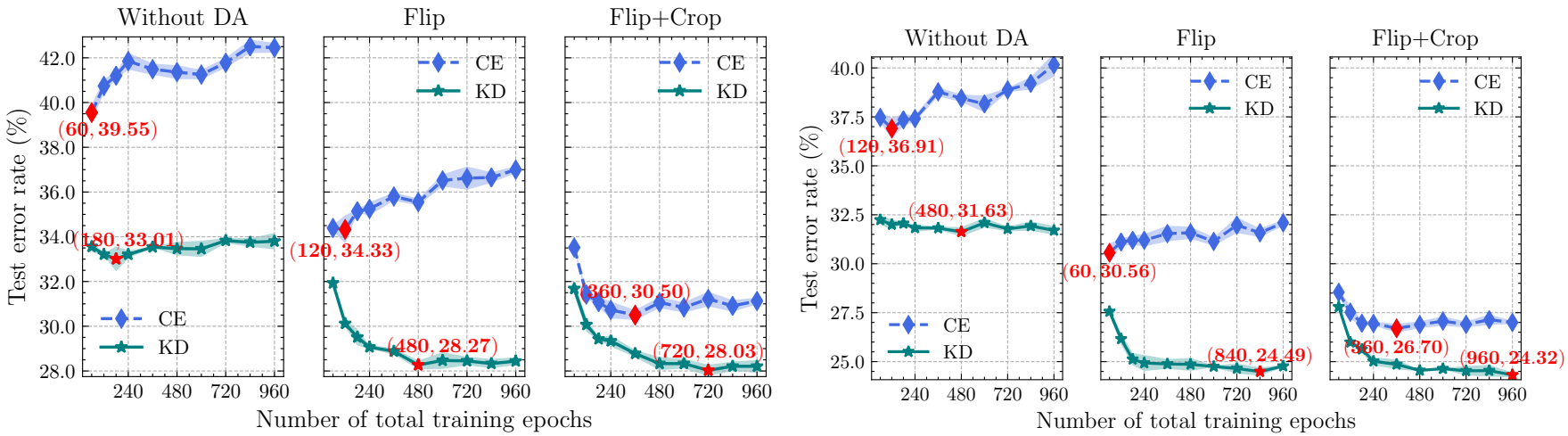
Figure: Illustration of KD (left) and some DA samples (right).

Background & Motivation: KD and DA

- Most KD papers try to improve KD based on the output (or internal features) of the network, by using better KD loss functions, *e.g.*, CRD (Tian et al., 2020). Few papers have attempted to understand or improve KD from the input side – *Esp.*, how KD interacts with DA has not been well understood so far.
- We ask in this paper: **What makes a "Good" data augmentation in KD?**

Background & Motivation: Why This Question Matters?

- Theoretically, it can help us towards a better understanding about how data augmentation plays a role in KD.
- Practically, it can bring us considerable performance gains. Below we show test error rates using the standard cross-entropy (CE) loss *vs.* using KD loss. Notably, we can obtain considerable performance gains simply via *stronger DA and more training epochs*, as also noted by (Beyer et al., 2022).



(a) resnet56/resnet20 (b) wrn_40_2/wrn_16_2

Figure: Test error rate on CIFAR100 of using KD *vs.* CE loss when trained for different epochs and with different DA schemes. "Flip": random horizontal flip; "Crop": random crop. The optimal number of training epochs and its test error rate are highlighted in red.

Motivation: Our Goal in This Paper

- Wait! **What do you mean by "stronger DA"?** Stronger DA means lower student test error. For example, it is straightforward that Flip+Crop is stronger than Flip alone. How about other DA schemes, like Mixup (Zhang et al., 2018) *vs.* AutoAugment (Cubuk et al., 2019)? Which is stronger?
- No clue? We thus desire a principled way (*e.g.*, a concrete metric) to make the vague concept "stronger" exact. Presenting such a theoretically sound metric and empirically validating its effectiveness is the goal of this paper.

Theoretical Investigation

Prerequisites: Multi-Class Classification with KD

Given a training set $S = \{(x_n, y_n)\}_{n=1}^N \sim \mathcal{D}^N$, where $\mathcal{D}$ is the joint distribution for input-output random variable pair (x, y), the goal in multi-class classification is to pin down a predictor $\mathbf{f} : \mathcal{X} \to \mathbb{R}^C$ from a hypothesis class $\mathcal{H}$, where $\mathcal{X}$ is the input space and C refers to the number of classes. **Three risks** worth our attention:

- The predictor $\mathbf{f}$ is supposed to minimize the *true risk* defined on data distribution.
- Since the data distribution is unknown, what we optimize in practice is the *empirical risk* defined on training set.
- In KD, the one-hot target is replaced with softened teacher's probability, which gives us the *empirical distilled risk*.

$$R_{\mathcal{D}}(\mathbf{f}) \overset{\text{def}}{=} \mathop{\mathrm{E}}_{(x,y)\sim\mathcal{D}} [L(y, \mathbf{f}(x))] \quad \triangleright \texttt{true risk} \tag{1}$$

$$R_S(\mathbf{f}) \overset{\text{def}}{=} -\frac{1}{N} \sum_{n=1}^{N} \mathbf{e}_{y_n}^\top \log(\mathbf{f}(x_n)) \quad \triangleright \texttt{empirical risk} \tag{2}$$

$$\hat{R}_S(\mathbf{f}) \overset{\text{def}}{=} -\frac{1}{N} \sum_{n=1}^{N} \mathbf{p}^{(t)}(x_n)^\top \log(\mathbf{f}(x_n)) \quad \triangleright \texttt{empirical distilled risk} \tag{3}$$

What Makes a "Good" DA in KD?

Proposition

Given a bounded loss function and a fixed teacher model with the empirical distilled risk, for any predictor $\mathbf{f}$*, consider two sampled sequences* $S_1 \in \mathcal{D}^N$ *and* $S_2 \in \mathcal{D}^N$*, they are made up of* N *elements sampled from the same distribution* $\mathcal{D}$*,* **while not i.i.d.** *(especially when data augmentation is employed). If the elements in* S_1 *present a larger correlation than those in* S_2*, then the student's generalization gap trained on* S_1 *will be greater than that trained on* S_2*:*

$$\mathop{\mathrm{E}}_{S_1 \sim \mathcal{D}^N} \left[(\hat{R}_{S_1}(\mathbf{f}) - R_{\mathcal{D}}(\mathbf{f}))^2 \right] > \mathop{\mathrm{E}}_{S_2 \sim \mathcal{D}^N} \left[(\hat{R}_{S_2}(\mathbf{f}) - R_{\mathcal{D}}(\mathbf{f}))^2 \right]. \tag{4}$$

Key Part in the Proof: Examining the Covariance among Teacher-Student Cross-Entropy

We define $q(x_i) = -\mathbf{p}^{(t)}(x_i)^\top \log(\mathbf{f}(x_i))$ – *i.e.*, **teacher-student cross-entropy**. The student generalization gap boils down to the covariance of the teacher-student cross-entropy (highlighted in red), which we need to minimize.

$$
\begin{aligned}
\mathrm{Var}_S[\hat{R}_S(\mathbf{f})] &= \mathrm{Var}_S[\frac{1}{N}\sum_{i=1}^{N} q(x_i)] = \frac{1}{N^2}\mathrm{Cov}_S[\sum_{j=1}^{N} q(x_j), \sum_{k=1}^{N} q(x_k)] \\
&= \frac{1}{N^2}\Big(\sum_{i=1}^{N} \mathrm{Var}_{x_i}[q(x_i)] + 2 \sum_{1\le j<k\le N} \mathrm{Cov}_S[q(x_j), q(x_k)]\Big) \\
&= \frac{1}{N^2}\big(N \cdot \mathrm{Var}_x[q(x)] + 2 \sum_{1\le j<k\le N} \mathrm{Cov}_S[q(x_j), q(x_k)]\big) \\
&= \frac{1}{N}\mathrm{Var}_x[q(x)] + \frac{2}{N^2} \sum_{1\le j<k\le N} \mathrm{Cov}_S[q(x_j), q(x_k)].
\end{aligned}
\tag{5}
$$

(Please refer to our paper for the complete proof.)

Practical Metric: T. Stddev

- In our proposition, the predictor $\mathbf{f}$ (*i.e.*, the student model) can be any one. For practical use, we must pick one particular student as oracle to conduct the evaluation.
- A trick for simplification: We can use the teacher as student, which gives us:

$$\begin{aligned}\mathrm{Cov}[q(x_j), q(x_k)] &= \mathrm{Cov}[\mathbf{p}^{(t)}(x_j)^\top \log(\mathbf{f}(x_j)), \mathbf{p}^{(t)}(x_k)^\top \log(\mathbf{f}(x_k))] \\ &= \mathrm{Cov}[\mathbf{p}^{(t)}(x_j)^\top \log(\mathbf{p}^{(t)}(x_j)), \mathbf{p}^{(t)}(x_k)^\top \log(\mathbf{p}^{(t)}(x_k)), \\ &\propto \mathrm{Cov}[\mathbf{p}^{(t)}(x_j), \mathbf{p}^{(t)}(x_k)]\end{aligned} \tag{6}$$

In this case, we only need the covariance of $\mathbf{p}^{(t)}(x_j)$ (*i.e.*, the teacher's probability) to measure the "goodness" of a certain DA technique, no need for the student.

Next, we show how to estimate this covariance and introduce the proposed metric.

Practical Metric: T. Stddev (Cont'd)

- The covariance of $\mathbf{p}^{(t)}(x)$ is hard to estimate accurately by sampling because of the *limited observation* problem (see our Appendix for more details).
- Instead, we look at the variance of *the average* of $\mathbf{p}^{(t)}(x)$:

$$\mathrm{Var}(\frac{1}{N}\sum_{i=1}^{N}\mathbf{p}^{(t)}(x_i)) = \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}(\mathbf{p}^{(t)}(x_i)) \quad \triangleright \texttt{Same for all inputs}$$
$$+ \frac{2}{N^2}\sum_{1\le j<k\le N}\mathrm{Cov}(\mathbf{p}^{(t)}(x_j), \mathbf{p}^{(t)}(x_k)). \tag{7}$$

  As seen, because all the input sample x's obey the same distribution, all $\mathbf{p}^{(t)}(x)$'s obey the same distribution. Then, $\mathrm{Var}(\mathbf{p}^{(t)}(x_i))$ is the same for all x_i's. Therefore, comparing $\mathrm{Cov}(\mathbf{p}^{(t)}(x_j), \mathbf{p}^{(t)}(x_k))$ is equivalent to comparing the LHS.
- The LHS is more stable to estimate – governed by the LLN, its estimation will stabilize when N is large enough.

Practical Metric: T. Stddev (Cont'd)

- In our paper, $\frac{1}{N}\sum_{i=1}^{N}\mathbf{p}^{(t)}(x_i)$ is termed the *teacher's mean probability*.
- The estimation for $\mathrm{Var}(\frac{1}{N}\sum_{i=1}^{N}\mathbf{p}^{(t)}(x_i))$ gives us an vector $\mathbf{m}\in\mathbb{R}_{+}^{C}$, then we take the square root (*i.e.*, make it from variance to stddev) and average it over the C classes, which finally gives us a scalar $\bar{m}$ – this is the proposed metric, termed the *stddev of teacher's mean probability*, *shorted as T. stddev*.

$$\mathbf{m}=\mathrm{Var}_{S^*}(\mathbf{u}),\mathbf{m}\in\mathbb{R}_{+}^{C},\quad \bar{m}=\frac{1}{C}\sum_{i\in[C]}(\mathbf{m}_i)^{\frac{1}{2}},\bar{m}\in\mathbb{R}_{+}. \tag{8}$$

Enhancing CutMix: CutMixPick

- Based on the proposed theory, we present an entropy-based data picking scheme to enhance CutMix, giving us a new proposed DA scheme called *CutMixPick*.
- We pick *harder examples* from an augmented batch for the student's learning.
- By "harder", we mean larger Shannon's Entropy,

$$H(\mathbf{p}^{(t)}(x)) = -\mathbf{p}^{(t)}(x)^{\top} \log(\mathbf{p}^{(t)}(x)). \tag{9}$$

- We will show this new DA delivers the *lowest* T. stddev and therefore the *best* performance on CIFAR100 and Tiny ImageNet.

Experimental Results

Data Augmentation Schemes in our Experiments

We investigate the following popular 7 DA schemes. Plus two we proposed, there would be 9 DA schemes in our experiments.

- **Identity**: This augmentation scheme simply makes a copy of each batch data.
- **Flip**: Random horizontal flip.
- **Flip+Crop**: Random horizontal flip and random crop. This is the standard DA extensively used in image classification (*e.g.*, on CIFAR and ImageNet).
- **Cutout** (DeVries & Taylor, 2017): Cutout occludes a small random patch of an image.
- **AutoAugment** (Cubuk et al., 2019): AutoAugment is an ensemble of a collected DA schemes. The DA policy is automatically selected by reinforcement learning instead of manually.
- **Mixup** (Zhang et al., 2018): Mixup applies linear interpolation between two inputs and applies the same linear interpolation to their labels to make the new label for the augmented sample.
- **CutMix** (Yun et al., 2019): CutMix cuts a small patch from a source image and pastes it to another source image. The resulted image is considered as a new input. The target for the new input is a linear interpolation from the two source labels.

Empirical Validation of Our Proposition

The proposition indicates that *a higher T. stddev should lead to a higher S. test loss*. Is this true in practice? **Yes!**

We plot the scatter points of S. test loss *vs.* T. stddev on the 9 DA schemes with various pairs on CIFAR100 and Tiny ImageNet. In all of them, S. test loss poses a strong correlation with T. stddev.

Empirical Validation of Our Proposition (CIFAR100)

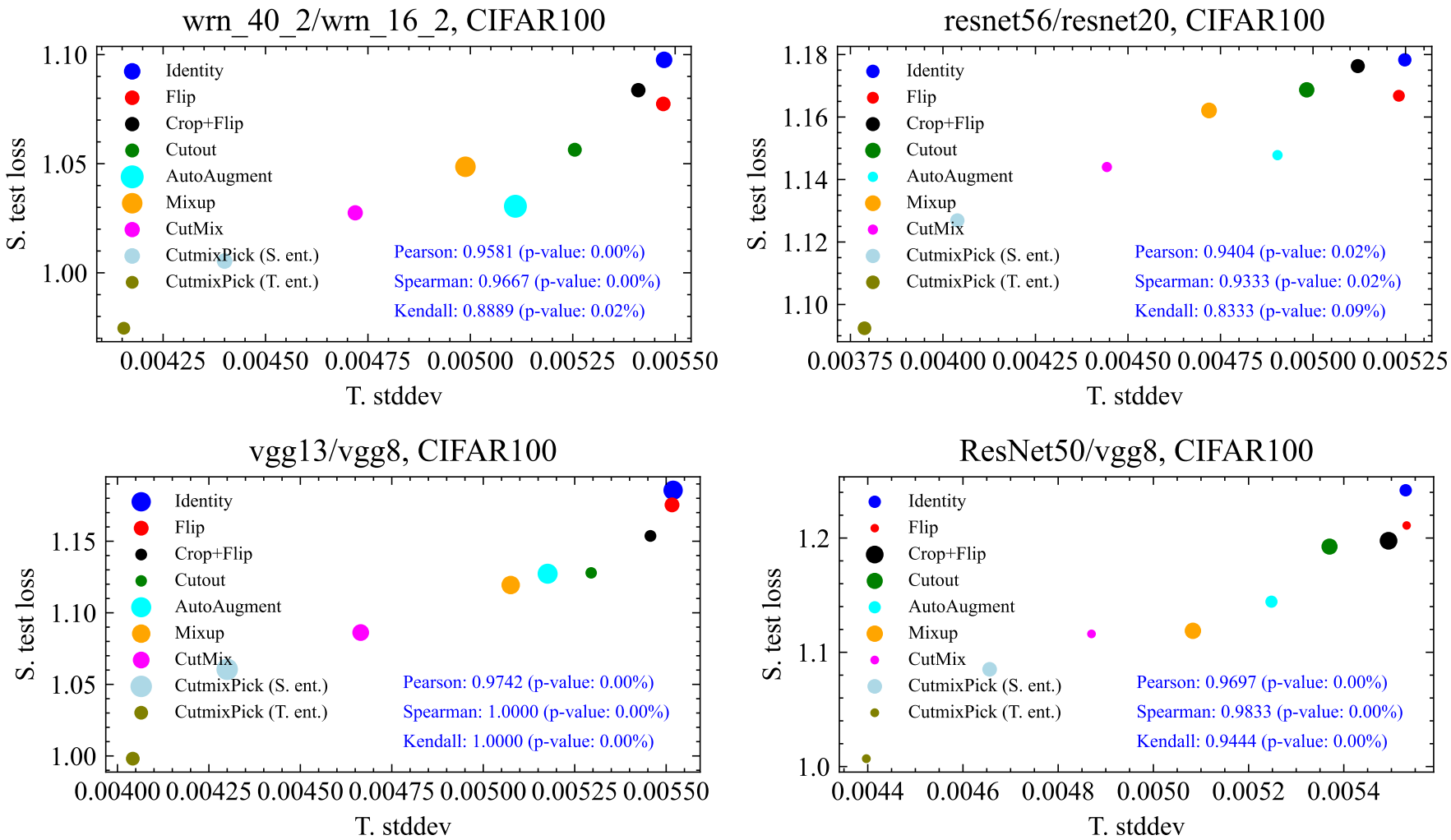


Figure: Scatter plots of T. stddev *vs.* S. test loss of different pairs on **CIFAR100**.

Empirical Validation of Our Proposition (Tiny ImageNet)

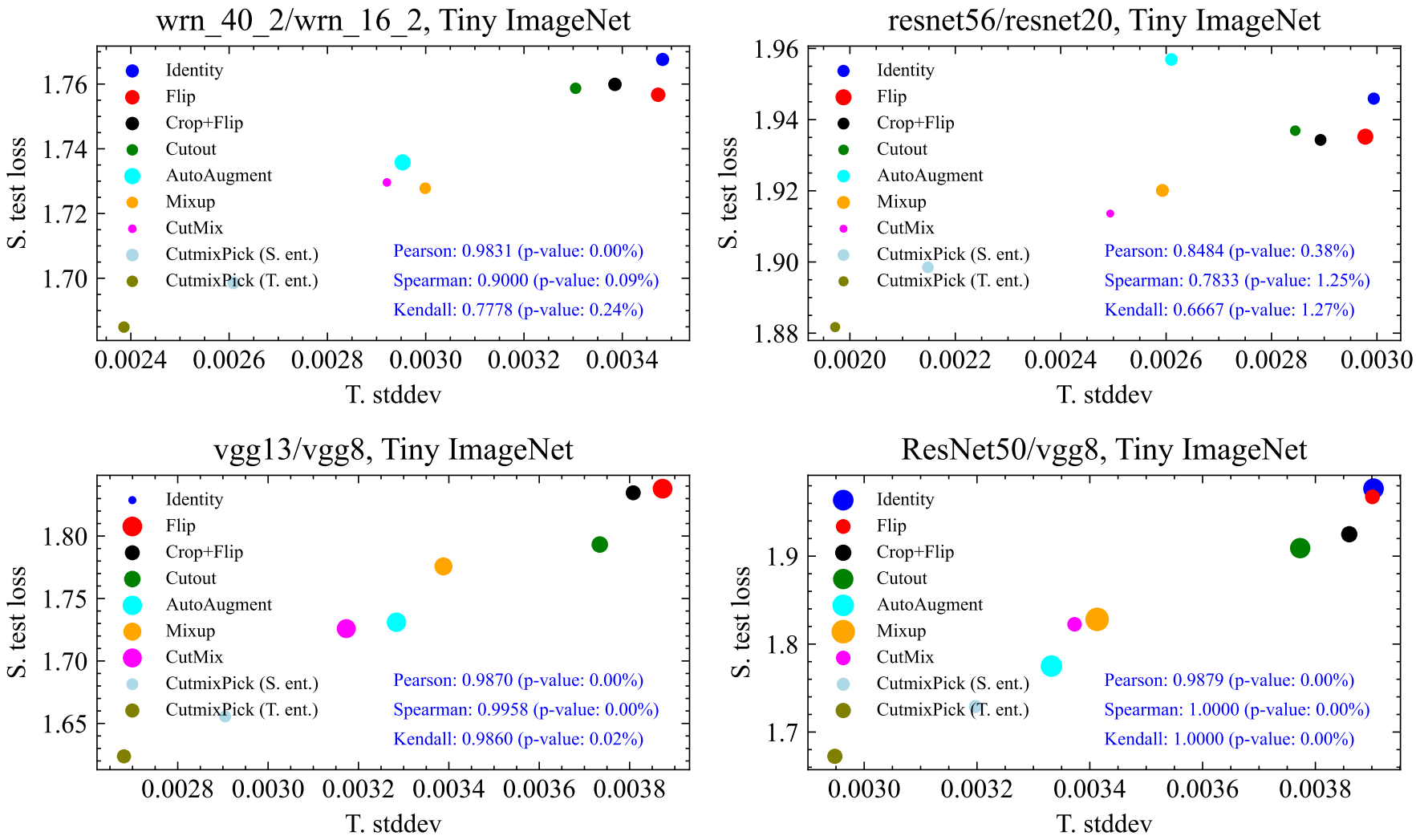


Figure: Scatter plots of T. stddev *vs.* S. test loss of different pairs on **Tiny ImageNet**.

How the Theory can Contribute in Practice?

We can easily harvest *considerable performance gains* simply by using a stronger DA with more training epochs, with the original KD loss function, shown below.

Teacher Student	wrn_40_2 wrn_16_2	resnet56 resnet20	resnet32x4 resnet8x4	vgg13 vgg8	vgg13 MobileNetV2	ResNet50 vgg8	resnet32x4 ShuffleV2
Teacher Acc.	75.61	72.34	79.42	74.64	74.64	79.34	79.42
Student Acc.	73.26	69.06	72.50	70.36	64.60	70.36	71.82
KD (Hinton et al., 2014)	74.92 (0.28)	70.66 (0.24)	73.33 (0.25)	72.98 (0.19)	67.37 (0.32)	73.81 (0.13)	74.45 (0.27)
KD+CutMix	75.34 (0.19)	70.77 (0.17)	74.91 (0.20)	74.16 (0.18)	68.79 (0.35)	74.85 (0.23)	76.61 (0.18)
KD+CutMixPick	75.59 (0.22)	70.99 (0.20)	74.78 (0.35)	74.43 (0.20)	69.49 (0.32)	74.95 (0.18)	76.90 (0.25)
KD_{960} (Hinton et al., 2014)	75.68 (0.12)	**71.79** (0.29)	73.14 (0.06)	74.00 (0.34)	68.77 (0.05)	74.04 (0.25)	74.64 (0.30)
KD+CutMixPick$_{960}$	**76.41** (0.10)	71.66 (0.15)	**75.12** (0.18)	**75.00** (0.17)	**70.47** (0.12)	**76.13** (0.16)	**77.90** (0.30)
CRD (Tian et al., 2020)	75.64 (0.21)	71.63 (0.15)	75.46 (0.25)	74.29 (0.12)	69.94 (0.05)	74.58 (0.27)	76.05 (0.09)
CRD+**CutMixPick**	75.96 (0.27)	71.41 (0.26)	**76.11** (0.53)	74.65 (0.12)	69.95 (0.22)	75.35 (0.22)	76.93 (0.11)
CRD+**CutMixPick$_{960}$**	**76.61** (0.01)	**72.40** (0.20)	75.96 (0.29)	**75.41** (0.10)	**70.84** (0.05)	**76.20** (0.22)	**78.51** (0.27)

Table: Student test accuracy on **CIFAR100**. Each result is obtained by 3 random runs, mean (std) accuracy reported. The best results are in **bold** and second best underlined. The subscript 960 means the total number of training epochs (default: 240).

Conclusion

Conclusion

- We present a proven proposition to precisely answer "*what makes a good data augmentation in knowledge distillation*": A good DA should reduce the variance (or covariance) of the teacher-student's cross-entropy.
- We present a practical useful metric that *only needs the teacher* to measure the "goodness" of a DA in KD: the *stddev of teacher's mean probability* (T. stddev).
- Interestingly, T. stddev works *very well* in practice (on CIFAR100 and Tiny ImageNet), posing a strong correlation with student's test loss, despite knowing *nothing* about the student.
- Based on the theory, we further propose an entropy-based data picking algorithm that can further boost prior SOTA DA scheme (CutMix) in KD, giving us a new stronger DA method, CutMixPick.
- Finally, we show how the theory can be utilized in practice to harvest considerable performance gains *simply by using a stronger DA and more training epochs*.

GitHub Code

Thank you!

Code: https://github.com/mingsun-tse/Good-DA-in-KD

References

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *CVPR*, 2022.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Workshop*, 2014.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *WACV*, 2019.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.