How and When Adversarial Robustness Improves in Knowledge Distillation?

Rulin Shao
Carnegie Mellon University
Pittsburgh, United States
rulins@cs.cmu.edu

Jinfeng Yi
JD AI Research
Shanghai, China
yijinfeng@jd.com

Cho-Jui Hsieh
University of California, Los Angeles
Los Angeles, United States
chohsieh@cs.ucla.edu

Pin-Yu Chen
IBM Research
New York, United States
pin-yu.chen@ibm.com

Abstract

Current works on knowledge distillation (KD) mainly focus on preserving the accuracy. However, other essential model properties, such as adversarial robustness, can be lost during distillation. This paper studies how and when the adversarial robustness can be transferred or improved from a teacher model to a student model in KD. We show that standard KD fails to preserve adversarial robustness, and we propose KD with input gradient alignment (KDIGA) for remedy. Under certain assumptions, we prove that the student model using our proposed KDIGA can achieve at least the same certified robustness as the teacher model. We also propose using KDIGA in an iterative self-distillation (ISD) training scheme, which can achieve better standard accuracy and adversarial robustness than adversarial training (AT), without using AT or any pre-trained robust teacher. Our experiments contain a diverse set of teacher and student models evaluated on ImageNet and CIFAR-10 datasets, including CNNs and ViTs. Our analysis shows several novel insights that (1) With KDIGA and ISD, students can preserve or even exceed the adversarial robustness of the teacher model, even when their models have fundamentally different architectures; (2) KDIGA enables robustness transfer to pre-trained students, such as KD from an adversarially trained ResNet to a pre-trained ViT, without loss of clean accuracy; and (3) Our derived local linearity bounds for characterizing adversarial robustness in KD are consistent with the empirical results.

1. Introduction

Knowledge distillation (KD) [14] is a popular machine learning framework for teacher-student training. To illustrate the critical but overlooked failure mode of standard KD, in Figure1 we show that it cannot preserve the adversarial robustness of the teacher model, and propose to use input gradient alignment in KD (we name it KDIGA) and iterative self-distillation (ISD) for improving/preserving both standard and robust accuracy. In addition to empirical evidence, in this paper, we also prove that our method can make the student achieve at least the same certified robustness as the teacher model under certain assumptions. When comparing our method with other baselines on ImageNet and CIFAR-10 datasets, the results show substantial improvement in the adversarial robustness of the student models obtained by our method.

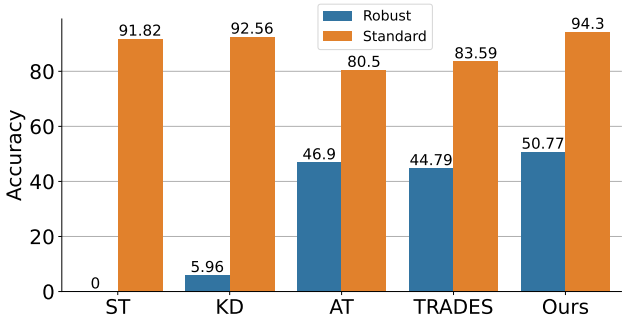

Figure 1. Clean accuracy (%) and robust accuracy (%) of MobileNetV2 obtained by different methods against 20-step PGD attack with an attack radius of 8/255 on CIFAR-10. "ST" stands for standard training, "KD" stands for knowledge distillation, "AT" and "TRADES" are adversarial training methods. (Scores are lower than their best reported because we use MobileNetV2 for comparison. See Table 1 for details.) KD uses WideResNet trained with TRADES as the teacher. Our method does not require robust teacher or adversarial training.

To demonstrate the generality of our proposed method, we further study the transferability of adversarial robustness

between convolutional neural networks (CNNs) and vision transformers (ViTs). We show that our method enables the transfer of adversarial robustness between these two fundamentally different architectures, and it can improve the adversarial robustness of a pre-trained ViT without sacrificing clean accuracy. We also extend our theoretical analysis and use local linearity measures to characterize the transfer of adversarial robustness in KD, and show that our derived performance bounds match the trends of the empirical robustness.

Inspired by the theoretical guarantee that the student can achieve at least the same certified robustness as the teacher in certain conditions, we further apply KDIGA with iterative self-distillation (ISD) training, which **boosts both the clean accuracy and the robust accuracy even without a pre-trained robust teacher, nor adversarial training**. Figure 1 shows our model significantly outperforms adversarially trained models.

Our Contributions:

- We propose to use KD with input gradient alignment (KDIGA) to train both accurate and adversarially robust student models. For instance, using KDIGA on CIFAR-10/ImageNet, the robust accuracy of the student model can be significantly increased from 5.97%→25.35%/1.5%→37.5% compared with KD, while simultaneously achieving even better clean accuracy.
- We propose a novel and efficient training scheme that combines KDIGA with iterative self-distillation (ISD), which achieves clean accuracy of 94.43% and robust accuracy of 50.77% against 20-step PGD attack on CIFAR-10 using a small model (MobileNetV2), without requiring any robust teacher, nor adversarial training.
- We show that adversarial robustness can be transferred between fundamentally different architectures with KDIGA, e.g. ResNet18 and ViTs.
- We prove that the student model distilled with KDIGA can achieve at least the same certified robustness as the teacher with some mild assumptions, and provide a bound for characterizing adversarial robustness in KD, which is consistent with the empirical results.

2. Robust Student Training

2.1. Knowledge Distillation with Input Gradient Alignment (KDIGA)

Suppose $f^s(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^N$ is the student model and $f^t(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^N$ is the teacher model, where D is the input dimension and N is the number of classes. Standard knowledge distillation aims to train the student model f^s that matches the logits of teacher model f^t, which won't provide any robustness guarantees both empirically and theoretically. Intuitively, the gradient with respect to input pixels captures how the prediction changes under small perturbation, which is directly related to the definition of robustness. Therefore, in KDIGA, we force the student to learn both the logits and gradient knowledge from the teacher model with the objective:

$$\arg\min_{f^s} \mathcal{L}_{IGA}(f^s; \boldsymbol{x}, y, f^t),$$

where $(\boldsymbol{x}, y) \in \mathcal{D}$ is the input image and the corresponding label in the training dataset, and

$$\begin{aligned}\mathcal{L}_{IGA} = \lambda_{CE}\mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) + \lambda_{KL}T^2\mathcal{L}_{KL}(f^s(\boldsymbol{x})/T, f^t(\boldsymbol{x})/T) \\ +\lambda_{IGA}\|\nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\|_2,\end{aligned}$$

where $\mathcal{L}_{CE}, \mathcal{L}_{KL}$ stand for cross-entropy loss and KL-divergence loss, $\lambda_{CE}, \lambda_{KL}$ and λ_{IGA} are scalar values, T is the temperature factor, and $\|\cdot\|_2$ is the ℓ_2 norm. Pseudo code of KDIGA is shown in Alg. 1 in Appendix G.

2.2. Iterative Self-distillation (ISD)

Under certain assumptions, we prove in Section 2.3 that the student's adversarial robustness can be at least as the teacher's when applying KDIGA in the ideal situation. This inspires us to iteratively boost the adversarial robustness using KDIGA combined with ISD:

$$\begin{cases} f_l^s = \arg\min_{f^s} \mathcal{L}_{IGA}(\text{Dropout}(f^s); \boldsymbol{x}, y, f_l^t) \\ f_{l+1}^t = f_l^s \end{cases}$$

where $l = 0, 1, \ldots, L$ stands for the iteration index. Dropout$(\cdot)$ means adding dropout to the last layer [27], which we found to be useful for avoiding getting stuck in the bad local minimum.

We consider the following two settings to choose the initial teacher f_0^t for ISD:

Without initial robust teacher (WoT): Train f_0^t from scratch using standard training with cross-entropy loss:

$$f_0^t = \arg\min_{f^s} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y). \tag{1}$$

Then ISD starts by treating f_0^t as the initial teacher.

With initial robust teacher (WiT): In the WiT setting, a pre-trained robust teacher is loaded as the initial teacher f_0^t. The robust teacher can be obtained from adversarial training with different model architectures. The ISD starts from distilling from f_0^t in the first loop and then conducts self-distillation in the following loops.

The pseudocode of ISD in WoT and WiT settings are shown in Algorithm 2 and Algorithm 3 respectively in Appendix G.

2.3. Preservation of Certified Robustness

In this section, we prove that using KDIGA, the student model can provably achieve as good robustness as the teacher model's in ideal situations.

Definition 1. *(ϵ-robust) Classifier $f(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^N$ is ϵ-robust if*

$$\arg\max f(\boldsymbol{x}+\boldsymbol{\delta}) = \arg\max f(\boldsymbol{x}),\ \forall x \in \mathcal{D}, \forall \delta \in [\mathbf{0}, \boldsymbol{\epsilon}]^D.$$

Under mild assumptions, we aim to show that if the teacher model has a robust radius of ϵ, then the student model is at least ϵ-robust under ideal situations. The first assumption is the *perfect student* assumption in which we suppose $f^s : \mathbb{R}^D \to \mathbb{R}^N$ is a student model distilled from the teacher model $f^t : \mathbb{R}^D \to \mathbb{R}^N$ using distillation loss $\mathcal{L}$, and f^s is a perfect student if $\mathcal{L}(\boldsymbol{x}, y) = 0, \forall (\boldsymbol{x}, y) \in \mathcal{D}$. The second assumption is *local linearity*, which assumes that neural networks with piece-wise linear activation functions are locally linear ([4, 15, 22]) and the certified robust area falls into these piece-wise linear regions. These two assumptions collaboratively build an ideal situation of knowledge distillation in which we can derive a strong property of KDIGA that the certified robustness of the student model can be as good as or even better than that of the teacher model. Proposition 1 concludes our statement.

Proposition 1. *Suppose the teacher model $f^t : \mathbb{R}^D \to \mathbb{R}^N$ is ϵ-robust, $f^s : \mathbb{R}^D \to \mathbb{R}^N$ is a perfect student trained using KDIGA, then f^s is at least ϵ-robust.*

We give a proof for Proposition 1 and illustrate why the knowledge distillation without input gradient alignment cannot preserve the adversarial robustness in Appendix A.

2.4. General Bound for the Adversarial Robustness of the Student Model

In this section, we derive a general bound for the adversarial robustness of the student model in KD. No assumption is needed for this bound and the result applies to any KD method. To derive this bound, we first introduce the Local Linearity Measure (LLM, [20]) in Definition 2. The proof for Proposition 2 can be found in Appendix B.

Definition 2. *(Local Linearity Measure) The local linearity of a classifier $f(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^N$ is measured by the maximum absolute difference between the cross-entropy loss and its first-order Taylor expansion in the ϵ-neighborhood:*

$$LLM(f, \boldsymbol{x}, \epsilon) = \max_{\boldsymbol{\delta} \in B(\epsilon)} \left| \mathcal{L}_{CE}(f(\boldsymbol{x}+\boldsymbol{\delta}) - \mathcal{L}_{CE}(f(\boldsymbol{x})) - \boldsymbol{\delta}^T \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f(\boldsymbol{x})) \right|$$

Proposition 2. *Consider a student model $f^s : \mathbb{R}^D \to \mathbb{R}^N$ distilled from a teacher model $f^t : \mathbb{R}^D \to \mathbb{R}^N$, then $\forall \boldsymbol{\delta} \in B(\epsilon)$,*

$$\left| \mathcal{L}_{CE}(f^s(\boldsymbol{x}+\boldsymbol{\delta}), y) - \mathcal{L}_{CE}(f^t(\boldsymbol{x}+\boldsymbol{\delta}), y) \right| \leq \gamma^s + \gamma^t + \phi$$

where $\gamma^s = LLM(f^s, \boldsymbol{x}, \epsilon)$, $\gamma^t = LLM(f^t, \boldsymbol{x}, \epsilon)$, and $\phi = \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) + \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y) + \epsilon \| \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y) \|$, and $\| \cdot \|$ is a norm.

Table 1. Robust accuracy (%) against 20-step PGD attack of 8/255 radius and clean accuracy (%) on the CIFAR-10 dataset. WRN(PGD-7)‡ is trained on MNIST, and WRN(PGD-7)† on TinyImageNet. ISD with "*" doesn't add dropout to the last layer. "Adv" indicates whether adversarial training is used.

Model	#Params	Adv	Clean	Robust
MNV2 (ST)	2M	N	91.82	0.00
WRN (TRADES) $\xrightarrow{\text{KD}}$ MNV2	2M	N	92.56	5.97
WRN (TRADES) $\xrightarrow{\text{ARD}}$ MNV2	2M	Y	91.65	20.73
WRN (PGD-7)† $\xrightarrow{\text{IGAM}}$ WRN	48M	Y	93.20	32.40
WRN (PGD-7)‡ $\xrightarrow{\text{IGAM}}$ WRN	48M	Y	93.60	43.50
WRN (PGD-7)	48M	Y	87.25	**45.90**
WRN (TRADES)	48M	Y	84.92	**56.68**
MNV2 (PGD-7)	2M	Y	80.50	**46.90**
MNV2 (TRADES)	2M	Y	83.59	**44.79**
WRN (TRADES) $\xrightarrow{\text{KDIGA}}$ MNV2	2M	N	93.03	25.35
WRN (TRADES) $\xrightarrow{\text{KDIGA-ARD}_\text{C}}$ MNV2	2M	Y	92.22	25.85
WRN (TRADES) $\xrightarrow{\text{KDIGA-ARD}_\text{A}}$ MNV2	2M	Y	90.67	27.50
WRN (TRADES) $\xrightarrow{\text{ISD-WiT-5}^*}$ MNV2	2M	N	**94.14**	**44.42**
MNV2 (ISD-WoT-1)	2M	N	92.68	32.98
MNV2 (ISD-WoT-2)	2M	N	**94.18**	34.77
MNV2 (ISD-WoT-3)	2M	N	**94.08**	**45.76**
MNV2 (ISD-WoT-4)	2M	N	**93.84**	**48.05**
MNV2 (ISD-WoT-5)	2M	N	**94.00**	**49.74**
MNV2 (ISD-WoT-6)	2M	N	**94.43**	**50.77**

3. Experiments

In this section, we evaluate our method on CIFAR-10 and ImageNet datsets. Details of settings can be found in Appendix D. We compare our method with baselines on CIFAR-10 and ImageNet with the corresponding results shown in Table 1 and Table 2 respectively. For fair comparison, we consider the settings where ARD (with adversarial training) and IGAM (with both adversarial training and an additional discriminator) can achieve a clean accuracy above 90%. For the experiments on ImageNet, the ARD results are omitted because it is difficult to generalize to large-scale datasets, which shows no convergence on ImageNet with a very low training speed, and no result for ImageNet is provided in [1]. Due to the limitation of computing resources, we ran all experiments on ImageNet for 50 epochs and remark that better performance could be achieved with more training epochs or ISD training loops. We conclude our key findings as below:

Standard KD Cannot Preserve Adversarial Robustness. As shown in Table 1, models trained using ST or KD are vulnerable to adversarial perturbations. The standard knowledge distillation can hardly preserve the adversarial robustness from teacher models The robust accuracy of KD is 5.97% and the robust accuracy of ST is 0%.

KDIGA and ISD Improve Adversarial Robustness without Sacrificing Clean Accuracy. As shown in Table 1, both KDIGA and ISD achieve clean accuracy above 93%, which is even higher than ST and KD. ISD has the highest clean accuracy due to iterative self-distillation, where the

Table 2. Robust accuracy (%) against 40-step PGD attack and clean accuracy (%) on the ImageNet dataset. Robust accuracy of the teacher models is shown in brackets.

Model	Clean	PGD Attack radius 0.001	0.01
ResNet18 (ST)	68.7 (-)	24.9 (-)	0.0 (-)
ViT-S/16 (ST)	77.6 (-)	55.4 (-)	1.0 (-)
ViT-S/16 (ST) $\xrightarrow{\text{KD}}$ ResNet18	69.0 (77.6)	30.1 (55.4)	0.0 (1.0)
ViT-S/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	60.0 (77.6)	51.0 (55.4)	3.3 (1.0)
ViT-B/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	64.7 (76.3)	52.8 (48.9)	0.7 (0.9)
ViT-L/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	65.9 (80.0)	53.2 (55.1)	1.4 (1.8)
DEIT-S/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	63.6 (77.7)	53.1 (48.9)	1.6 (1.1)
ResNet50 (AT) $\xrightarrow{\text{KD}}$ ResNet18	66.3 (63.1)	25.7 (61.9)	0.0 (49.0)
ResNet50 (AT) $\xrightarrow{\text{KDIGA}}$ ResNet18	54.2 (63.1)	48.2 (61.9)	9.2 (49.0)
ResNet50 (AT) $\xrightarrow{\text{KDIGA}}$ ResNet34	59.2 (63.1)	53.9 (61.9)	12.1 (49.0)
ResNet50 (AT) $\xrightarrow{\text{KDIGA}}$ ResNet50	58.8 (63.1)	53.7 (61.9)	12.4 (49.0)
ResNet50 (AT) $\xrightarrow{\text{KDIGA}}$ ResNet101	60.3 (63.1)	55.3 (61.9)	12.7 (49.0)
ResNet50 (AT) $\xrightarrow{\text{KDIGA}}$ ViT-S/16* [1]	77.7 (63.1)	65.3 (61.9)	11.1 (49.0)

clean accuracy is further improved with additional loops. Adversarial training can also achieve high adversarial robustness but at the cost of clean accuracy.

Dropout Can Help Stabilize the Self-Distillation in ISD and Avoid Getting Stuck in the Local Minimum. We show in Appendix F that, without dropout in the last layer, ISD can fall into bad local minimum and get stuck in the following self-distillation loops.

KDIGA Can Scale-Up to Large-Scale Datasets We show in Table 2 that KDIGA still works for ImageNet and show better preservation of adversarial robustness compared with KD. When distilling ResNet18 from ResNet50 (AT) and testing against 40-step PGD with a radius of 0.003, KDIGA has a robust accuracy of 37.5% in comparison with 1.5% for KD and 2.0% for ST. And it's the same for ViT-S/16.

Adversarial Robustness Can Transfer Between CNNs and Vision Transformers. According to Table 2, the robustness of ViT got improved with KDIGA, indicating we could transfer adversarial robustness from smaller CNNs to ViTs.

Input gradient alignment Works for Pre-trained Models. In the experiments of "ResNet50 $\xrightarrow{KDIGA}$ ViT-S/16*" as shown in Table 2, we take the pre-trained ViT as the student to help the training converge in a shorter time. This result shows the feasibility to further improve the adversarial robustness of a pre-trained model using KDIGA without harming the clean accuracy, which gives a novel and inspiring approach to train new robust models more efficiently at less or no cost of the clean accuracy.

Students Can Obtain Even Better Adversarial Robustness than Teachers. When distilling from DEIT-S/16 to ResNet18 with KDIGA achieves robust accuracy of 53.1% while the teacher model only has 48.9% in the same situations.

[1]The pre-trained student model is denoted with "*" where the distillation is conducted as fine-tuning.

Table 3. Bounds for adversarial robustness on CIFAR-10. llm_ϵ is defined by Definition 2 where ϵ is the radius of perturbations. l_{CE} is the cross-entropy loss. $\|g^s - g^t\|_2$ calculates the l_2-norm of input gradient alignment term with the same teacher.

Model	$llm_{4/255}$	$llm_{8/255}$	l_{CE}	$\|g^s - g^t\|_2$
MNV2 (ST)	12.413	21.691	0.364	4.099
WRN(TRADES) $\xrightarrow{\text{KD}}$ MNV2	5.960	10.286	**0.218**	1.958
WRN(TRADES) $\xrightarrow{\text{ARD}}$ MNV2	1.326	3.034	0.261	0.569
WRN(TRADES) $\xrightarrow{\text{KDIGA}}$ MNV2	2.561	4.914	0.235	0.587
WRN(TRADES) $\xrightarrow{\text{KDIGA-ARD}_\text{C}}$ MNV2	**1.081**	**2.421**	0.228	**0.339**
WRN(TRADES) $\xrightarrow{\text{KDIGA-ARD}_\text{A}}$ MNV2	1.107	2.442	0.285	0.377

3.1. Local Linearity Bounds for Adversarial Robustness in Knowledge Distillation

Table 3 shows the bounds for adversarial robustness of models trained on CIFAR-10. We randomly sample 1000 test samples to calculate the terms in the bounds. In reference to Table 1 and Table 3, the empirical performance matches the theoretical insights that models with better adversarial robustness have smaller values in the bounds. Table 3 also shows that combining our method with ARD can further reduce the bounds and induce better adversarial robustness. KD only has the lowest cross-entropy loss while other terms are high, which can explain its failure in preserving adversarial robustness, as its objective design only focuses on improving standard accuracy.

References

[1] Alvin Chan, Yi Tay, and Yew-Soon Ong. What it thinks is important is important: Robustness transfers through input gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 332–341, 2020. 3, 8

[2] Alvin Chan, Yi Tay, and Yew-Soon Ong. What it thinks is important is important: Robustness transfers through input gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 332–341, 2020. 7

[3] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 699–708, 2020. 7

[4] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In *the 22nd International Conference on Artificial Intelligence and Statistics*, pages 2057–2066. PMLR, 2019. 3

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse

parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020. 8

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 8

[8] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. 8

[9] Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *NeurIPS*, 2021. 7

[10] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3996–4003, 2020. 8, 10

[11] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34, 2021. 10

[12] 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. 8

[13] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019. 7

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[15] Guang-He Lee, David Alvarez-Melis, and Tommi S Jaakkola. Towards robust, locally linear deep networks. *arXiv preprint arXiv:1907.03207*, 2019. 3

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 8

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 8

[18] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021. 8

[19] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2021. 8

[20] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *arXiv preprint arXiv:1907.02610*, 2019. 3

[21] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 8

[22] Ben Sattelberg, Renzo Cavalieri, Michael Kirby, Chris Peterson, and Ross Beveridge. Locally linear attributes of relu neural networks. *arXiv preprint arXiv:2012.01940*, 2020. 3

[23] Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. *arXiv preprint arXiv:1905.08232*, 2019. 7

[24] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *arXiv preprint arXiv:2103.15670*, 2021. 8

[25] Ren Wang, Kaidi Xu, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Chuang Gan, and Meng Wang. On fast adversarial robustness adaptation in model-agnostic meta-learning. In *International Conference on Learning Representations*, 2021. 7

[26] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 8

[27] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 2, 8

[28] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 8

[29] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. 8

Appendix

A. Proof for Proposition 1

We note that the notation of δ and ϵ are reversed in this proof compared with the use in the main text.

Since f^t is δ-robust, the prediction of $f^t(\boldsymbol{x})$ is invariant to the input perturbations smaller than the certified robust radius by definition, i.e.,

$$\arg\max\ f^t(\boldsymbol{x}+\boldsymbol{\epsilon}) = \arg\max\ f^t(\boldsymbol{x}),\ \ \forall \boldsymbol{x} \in \mathcal{D},\ \ \forall \boldsymbol{\epsilon} \in (0,\delta)^D, \tag{2}$$

where $\mathcal{D}$ is the task-specific data set. Denote the student model distilled from the teacher model using normal knowledge distillation as $f^{KD}(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^N$. The loss of the normal knowledge distillation can be formulated as

$$\mathcal{L}_{KD}(\boldsymbol{x}, y) = \lambda_{CE}\mathcal{L}_{CE}(f^{KD}(\boldsymbol{x}), y) + \lambda_{KL}T^2\mathcal{L}_{KL}(f^{KD}(\boldsymbol{x})/T, f^t(\boldsymbol{x})/T),\ \ \forall (\boldsymbol{x}, y) \in \mathbb{D}, \tag{3}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, $\mathcal{L}_{KL}$ is the KL-divergence loss which is also called the soft loss in knowledge distillation, T is the temperature factor, and $\lambda_{CE}, \lambda_{KL}$ are hyper-parameters to balance the effects of the two losses. The loss of KDIGA is calculated by

$$\begin{aligned}\mathcal{L}_{IGA}(\boldsymbol{x}, y) =& \lambda_{CE}\mathcal{L}_{CE}(f^{IGA}(\boldsymbol{x}), y) + \lambda_{KL}T^2\mathcal{L}_{KL}(f^{IGA}(\boldsymbol{x})/T, f^t(\boldsymbol{x})/T) \\ &+ \lambda_{IGA}\|\nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^{IGA}(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\|_2,\ \ \forall (\boldsymbol{x}, y) \in \mathbb{D},\end{aligned} \tag{4}$$

where f^{IGA} is the student model, $\lambda_{CE}, \lambda_{KL}$ and λ_{IGA} are hyper-parameters.

Without loss of generality, we set the temperature factor $T = 1$ for both KD and KDIGA. According to the perfect student assumption, f^{IGA} satisfies the following equations:

$$\begin{cases} \nabla_{\boldsymbol{x}}\mathcal{L}_{IGA}(\boldsymbol{x}, y) - \nabla_{\boldsymbol{x}}\mathcal{L}_{IGA}(\boldsymbol{x}, y) = 0 & (5) \\ f^{IGA}(\boldsymbol{x}) - f^t(\boldsymbol{x}) = 0 & (6) \\ f^{IGA}(\boldsymbol{x}) = y, \qquad \forall (\boldsymbol{x}, y) \in \mathcal{D}. & (7) \end{cases}$$

The cross-entropy loss is defined as

$$\mathcal{L}_{CE}(f(\boldsymbol{x}), y) = -\log\big(\frac{\exp(f(\boldsymbol{x})_y)}{\sum_j \exp(f(\boldsymbol{x})_j)}\big) = -f(\boldsymbol{x})_y + \log(\sum_j \exp(f(\boldsymbol{x})_j)), \tag{8}$$

where $f(\cdot)$ is a classifier and $f(\boldsymbol{x})_j$ is the j-th prediction of the output. Then the gradient of the cross-entropy loss with respect to the input is

$$\begin{aligned}\nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f(\boldsymbol{x}), y) &= -\nabla_{\boldsymbol{x}} f(\boldsymbol{x})_y + \nabla_{\boldsymbol{x}} \log(\sum_j \exp(f(\boldsymbol{x})_j)) \\ &= -\nabla_{\boldsymbol{x}} f(\boldsymbol{x})_y + \frac{\nabla_{\boldsymbol{x}}(\sum_i \exp(f(\boldsymbol{x})_i))}{\sum_j \exp(f(\boldsymbol{x})_j)} \\ &= -\nabla_{\boldsymbol{x}} f(\boldsymbol{x})_y + \frac{\sum_i \nabla_{\boldsymbol{x}} \exp(f(\boldsymbol{x})_i)}{\sum_j \exp(f(\boldsymbol{x})_j)} \\ &= -\nabla_{\boldsymbol{x}} f(\boldsymbol{x})_y + \frac{\sum_i \exp(f(\boldsymbol{x})_i)\nabla_{\boldsymbol{x}} f(\boldsymbol{x})_i}{\sum_j \exp(f(\boldsymbol{x})_j)}\end{aligned} \tag{9}$$

Denote $\boldsymbol{g} = g(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$, $\boldsymbol{\alpha} = \text{softmax}(f(\boldsymbol{x}))$, then

$$\begin{aligned}\nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f(\boldsymbol{x}), y) &= -g(\boldsymbol{x})_y + \frac{\sum_i \exp(f(\boldsymbol{x})_i) g(\boldsymbol{x})_i}{\sum_j \exp(f_j(\boldsymbol{x}))} \\ &= -g(\boldsymbol{x})_y + \boldsymbol{\alpha} \cdot \boldsymbol{g} \\ &= (\boldsymbol{\alpha} - \boldsymbol{i}_y) \cdot \boldsymbol{g}.\end{aligned} \tag{10}$$

where $\boldsymbol{i}_y = (0, \cdots, 0, 1, 0, \cdots, 0)$ is an unit vector of which the y-th element equals one. According to Eq. 6, $\boldsymbol{\alpha}^t = \boldsymbol{\alpha}^{IGA} = \boldsymbol{\alpha}$. The third term in Eq. 4 for input gradient alignment is

$$\begin{aligned} &\|\nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^{IGA}(\boldsymbol{x}), y)\| \\ =&\|(\boldsymbol{\alpha}^t - \boldsymbol{i}_y) \cdot \boldsymbol{g}^t - (\boldsymbol{\alpha}^{IGA} - \boldsymbol{i}_y) \cdot \boldsymbol{g}^{IGA}\| \\ =&\|(\boldsymbol{\alpha} - \boldsymbol{i}_y) \cdot (\boldsymbol{g}^t - \boldsymbol{g}^{IGA})\|. \end{aligned} \tag{11}$$

Given $\boldsymbol{\alpha} - \boldsymbol{i}_y \neq \mathbf{0}$, $\boldsymbol{g}^t - \boldsymbol{g}^{IGA}$ must be $\mathbf{0}$ since $\boldsymbol{\alpha} - \boldsymbol{i}_y$ and $\boldsymbol{g}^t - \boldsymbol{g}^{IGA}$ are not strictly orthogonal unless $\boldsymbol{g}^t - \boldsymbol{g}^{IGA} = \mathbf{0}$. According to Eq. 5, we have $\boldsymbol{g}^t - \boldsymbol{g}^{IGA} = \mathbf{0}$.

According to the local linearity assumption,$\forall \boldsymbol{x} \in \mathbb{D}$, $\forall \boldsymbol{\epsilon} \in [0, \delta)^{H \times W \times C}$,

$$\begin{aligned} f^{IGA}(\boldsymbol{x} + \boldsymbol{\epsilon}) &= f^{IGA}(\boldsymbol{x}) + \boldsymbol{\epsilon}^T \cdot g^{IGA}(\boldsymbol{x}) \\ &= f^t(\boldsymbol{x}) + \boldsymbol{\epsilon}^T \cdot g^t(\boldsymbol{x}) \\ &= f^t(\boldsymbol{x} + \boldsymbol{\epsilon}) = f^t(\boldsymbol{x}) = f^{IGA}(\boldsymbol{x}). \end{aligned} \tag{12}$$

Therefore, the certified robust radius of f^{IGA} is at least δ, which proves Proposition 1.

However, the knowledge distillation without input gradient alignment cannot guarantee the adversarial robustness preservation. Suppose f^{KD} is a perfect student, we have

$$\begin{cases} f^{KD}(\boldsymbol{x}) - f^t(\boldsymbol{x}) = 0 & (13) \\ f^{KD}(\boldsymbol{x}) = y, \qquad \forall (\boldsymbol{x}, y) \in \mathcal{D}. & (14) \end{cases}$$

We point out that f^{KD} can have different predictions around $\boldsymbol{x}$, for example, let $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon} \in \mathring{\mathrm{B}}(\boldsymbol{x}, \delta)$, denote $h(\boldsymbol{x}) = f^{KD}(\boldsymbol{x}) - f^t(\boldsymbol{x})$, then $h(\boldsymbol{x}) = 0$, $\forall (\boldsymbol{x}, y) \in \mathcal{D}$ according to Eq. 13. But $\exists h(\boldsymbol{x})$, $\exists \boldsymbol{x} \in \mathring{\mathrm{B}}(\boldsymbol{x}, \delta)$ s.t.

$$\arg\max f^{KD}(\boldsymbol{x}) \neq \arg\max f^t(\boldsymbol{x}) \tag{15}$$

since the first-order derivative of $h(\boldsymbol{x})$ is not constrained to be 0 in the neighbourhood of $\boldsymbol{x}$. This means the predictions of the student model distilled using knowledge distillation without input gradient alignment can be altered if we add perturbations to the input image.

B. Proof for Proposition 2

We note that the notation of δ and ϵ are reversed in this proof compared with the use in the main text.

$$\begin{aligned} &\left|\mathcal{L}_{CE}(f^s(\boldsymbol{x} + \boldsymbol{\epsilon}), y) - \mathcal{L}_{CE}(f^t(\boldsymbol{x} + \boldsymbol{\epsilon}), y)\right| \\ =&\Big|\mathcal{L}_{CE}(f^s(\boldsymbol{x} + \boldsymbol{\epsilon}), y) - \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) \\ &- \left(\mathcal{L}_{CE}(f^t(\boldsymbol{x} + \boldsymbol{\epsilon}), y) - \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y) - \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\right) \\ &+ \left(\mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\right) \\ &+ \boldsymbol{\epsilon}^T \left(\nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\right) \Big| \\ \leq& \max_{\boldsymbol{\epsilon} \in B(\delta)} \left|\mathcal{L}_{CE}(f^s(\boldsymbol{x} + \boldsymbol{\epsilon}), y) - \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y)\right| \\ &+ \max_{\boldsymbol{\epsilon} \in B(\delta)} \left|\mathcal{L}_{CE}(f^t(\boldsymbol{x} + \boldsymbol{\epsilon}), y) - \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y) - \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\right| \\ &+ \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) + \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y) + \delta \|\nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\|. \end{aligned} \tag{16}$$

C. Related Work

There are some recent works studying when and how adversarial robustness will transfer in different machine learning settings, such as transfer learning [3, 13, 23], representation learning [2, 9] and Model-agnostic meta-learning (MAML) [25]. In contrast, we focus on the setting of knowledge distillation. The basic Knowledge Distillation (KD) formulates the supervised learning objective as

$$\arg\min_{f^s} \mathcal{L}_{KD}(\boldsymbol{x}, y) = \arg\min_{f^s} \lambda_{CE} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) + \lambda_{KD} T^2 \mathcal{L}_{KL}(f^s(\boldsymbol{x})/T, f^t(\boldsymbol{x})/T) \tag{17}$$

where f^s is the student model, f^t is the teacher model, $\boldsymbol{x}$ is a data sample and y is its label, $(\boldsymbol{x}, y) \in \mathcal{D}$, $\mathcal{D}$ is the training set, $\mathcal{L}_{CE}$ is the cross-entropy loss, $\mathcal{L}_{KL}$ is the KL-divergence loss, λ_{CE} and λ_{KD} are constant factors to balance the two losses, and T is a temperature factor. NoisyStudent proposed by [27] boosts the generalization performance of semi-supervised learning by training the model iteratively and reusing the student as the next-loop teacher. The authors in [1] study the robust transfers across different tasks and propose input gradient adversarial matching (IGAM). They train a student model that semantically resembles the teacher's input gradient with an additional discriminator network. In contrast, we focus on the adversarial robustness preservation and improvement in KD.

Projected gradient descent (PGD) is one of the most commonly used adversarial attacks for both adversarial robustness evaluation and adversarial training, which solves

$$\arg\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}_{CE}(f^s(\boldsymbol{x}+\delta), y) \tag{18}$$

by iteratively taking gradient ascent:

$$\mathbf{x}_{t+1}^{adv} = Clip_{\mathbf{x}_0,\epsilon}(\mathbf{x}_t^{adv} + \alpha \cdot \operatorname{sgn}\left(\nabla_{\mathbf{x}} \mathcal{L}_{CE}\left(\mathbf{x}_t^{adv}, y\right)\right)), \tag{19}$$

where $t = 1, \cdots, T$, T is the number of iterations, $\mathbf{x}_t^{adv}$ stands for the solution after t iterations, ∇_x denotes the gradient with respect to x, and $Clip_{\mathbf{x}_0,\epsilon}(\cdot)$ denotes clipping the values to make each $\mathbf{x}_{t+1}^{adv}$ within $[\mathbf{x}_0 - \epsilon, \mathbf{x}_0 + \epsilon]$, according to the ℓ_p norm bounded threat model. The adversarial perturbation is then obtained by $\delta_{\text{pgd}} = \mathbf{x}_T^{adv} - \mathbf{x}_0$. In addition, AutoAttack ([5]) is an ensemble of several adversarial attacks which evaluates adversarial robustness in a parameter-free manner. One effective way to train an adversarially robust model is adversarial training [8, 16, 29], which adds adversarial perturbations to the inputs during training and forces the model to learn robust predictions. [10] follows the same idea and formulates an adversarially robust distillation (ARD) objective using adversarial examples. However, it is computationally expensive to calculate the PGD adversarial perturbations which is unaffordable for large dataset like the ImageNet [6]. Besides, adversarial training reaches a high robust accuracy with a serious drop in sanity accuracy.

D. Settings

Teacher Models. We use pre-trained and publicly available neural networks of varying architectures as teacher models. For the CIFAR-10 dataset, we use the WideResNet [28] adversarially trained with TRADES following the setting in [29] as the teacher model. For the ImageNet dataset, we use both adversarially trained CNNs and normally trained vision transformers (ViTs) as the teacher models. We use the checkpoint of ResNet50 [12] provided by [8] which is adversarially trained with an attack radius of $4/255$. We also incorporate ViTs [7] as teacher models because they are shown to have better adversarial robustness than CNNs [18, 19, 24], and we are interested in the transferability of adversarial robustness between fundamentally different architectures, i.e. CNNs and ViTs.

Student Models. For the CIFAR-10 dataset, we use MobileNetV2 [21] as the student model. For the ImageNet dataset, we mainly use ResNet18 [12] as the student model for experiments . To study the effect of model size, we also consider ResNet34, ResNet50 and ResNet101. In addition, we use ViT-S/16 [7] as the student model to study the transferability of adversarial robustness from a CNN teacher to a ViT student. Unless specified, the student models are all trained from scratch. Because the training of ViT is difficult without large-scale pre-training [7], we use the pre-trained version provided by [26] and apply the knowledge distillation methods as a fine-tuning process.

Evaluation Metrics. Using the test sets of ImageNet and CIFAR-10, we report the best standard accuracy and the robust accuracy against adversarial attacks of the student models. We conduct ℓ_∞ norm bounded adversarial perturbations to generate adversarial examples for evaluating robust accuracy (the pixel value is scaled between 0 to 1), where we use a 40-step projected gradient descent (PGD) attack [16] and the parameter-free AutoAttack [5] for 1000 ImageNet test samples, and a 20-step PGD attack and AutoAttack for all CIFAR-10 test samples. Results of AutoAttack are supplemented in Appendix H.

Notation of Comparative Methods. We denote the standard knowledge distillation method as "KD", the method proposed by [1] as "IGAM", the method proposed by [10] as "ARD", our method without iterative self-distillation as "KDIGA", our method with iterative self-distillation as 'ISD-s' where s stands for the number of loops we run for self-distillation, and the two kinds of combinations of KDIGA and ARD defined in Appendix I as "KDIGA-ARD$_C$" and "KDIGA-ARD$_A$". We also compare our methods with two popular adversarial training techniques: training against a 7-step PGD adversary (PGD-7 [17]) and TRADES [29]. Unless otherwise stated, "ST" means the model is trained following the standard approach without distillation nor adversarial training, "TRADES" means the model is adversarially trained using TRADES, "PGD-7" means the model is adversarially trained against a 7-step PGD adversary, "MNV2" stands for MobileNetV2, and "WRN" stands

Table 4. Robust Accuracy (%) against 20-step PGD attack with an attack radius of 8/255.

Model	ISD-WoT-1	ISD-WoT-2	ISD-WoT-3
With Dropout	32.98	34.77	45.76
Without Dropout	24.32	32.69	41.11

for WideResNet. "Teacher $\xrightarrow{\text{Method}}$ Student" stands for the distillation from the "Teacher" to the "Student" using "Method". Training configurations can be found in Appendix E.

E. Training Configuration

For fair comparison, the coefficients of the cross-entropy loss and KL-divergence loss are both set to 0.5 for all distillation baselines. For experiments on CIFAR-10, we run KDIGA for 150 epochs with an initial learning rate of 0.1 with milestones at $[50, 100]$ of a decreasing rate of 0.1. The SGD optimizer with a momentum of 0.9 and a weight decay of 0.0002 is used to update the parameters. The coefficient for the input gradient alignment is $\frac{1}{B}$, where B is the batch size. We also evaluate the performance of ISD on CIFAR-10. We set the initial learning rate to 0.1 and the learning rate decay for fine-tuning to 0.01 as described in Section 2.2. For knowledge distillation on ImageNet, we run all distillation for 50 epochs with a batch size of 128, an initial learning rate of 0.1 for training from scratch and 0.00001 for fine-tuning, with milestones at $[20, 30, 40]$ of a decreasing rate of 0.1. The SGD optimizer with 0.9 momentum is used to update the model parameters, and a weight decay of 0.0001 is applied. The coefficient of the input gradient alignment term is $\frac{10^3}{B}$, where B is the batch size of the inputs. We use 1 Nvidia Quadro RTX 6000 to run the experiments on CIFAR-10 and 4 for ImageNet.

F. Dropout in ISD

To study the effect brought by the dropout layer in ISD, we conduct a ablation study as shown in Table 4. In an extreme situation, when apply ISD without dropout, the performance slightly drop in each loop, and we observe an early convergence to a lower robust accuracy when training without dropout.

G. Pseudocodes of KDIGA, ISD-WoT and ISD-WiT

Algorithm 1: Pseudocode of KDIGA

Input: teacher f^t, student f^s_θ with trainable parameters θ, training set $\mathcal{D}$, λ_{CE}, λ_{KL}, λ_{IGA}, learning rate η, # of epochs N_{epochs}
Output: adversarially robust student f^s_θ.
for *epoch* $\in N_{epochs}$ **do**
  for *batch* $(\boldsymbol{x}, y) \in \mathcal{D}$ **do**
    $p_s, p_t \leftarrow f^s_\theta(\boldsymbol{x}), f^t(\boldsymbol{x})$;
    $\ell_s, \ell_t \leftarrow \mathcal{L}_{CE}(p_s, y), \mathcal{L}_{CE}(p_t, y)$;
    $\ell_{KL} \leftarrow T^2 \mathcal{L}_{KL}(p_s/T, p_t/T)$;
    $g_s, g_t \leftarrow \nabla_{\boldsymbol{x}} \ell_s, \nabla_{\boldsymbol{x}} \ell_t$;
    $\ell_{iga} \leftarrow \lambda_{CE}\ell_s + \lambda_{KL}\ell_{KL} + \lambda_{IGA}\|g_s - g_t\|_2$;
    $\theta \leftarrow \theta - \eta \nabla_\theta \ell_{iga}$;

Algorithm 2: Pseudocode of ISD-WoT

Input: number of loops L, initial learning rate η, learning rate decay factor μ.
Output: adversarially robust student f^s_L.
$f^t_0 \leftarrow \arg\min_{f^s} \mathcal{L}_{CE}(f^s(\boldsymbol{x}), y)$ ◁ Train from scratch according to Eq. 1 with learning rate η
for $l \in \{1, \ldots, L\}$ **do**
  $f^s \leftarrow$ Dropout(f^s_{l-1}) ◁ Add dropout to the last layer
  $f^s_l \leftarrow \arg\min_{f^s} \mathcal{L}_{IGA}(f^s; \boldsymbol{x}, y, f^t_l)$ ◁ Fine-tune with learning rate $\mu\eta$ according to Algorithm 1
  $f^t_{l+1} \leftarrow f^s_l$

Table 5. Robust accuracy (%) of student models against AutoAttack with 8/255 on CIFAR-10. The robust teacher model is WideResNet [11]. We set the trade-off hyper-parameter for KD and KDIGA losses both to 0.5, and train for 15 epochs using KDIGA.

Model	Epoch	Clean	AA(8/255)
PreActResNet18	15	75.54	41.00

Algorithm 3: Pseudocode of ISD-WiT

Input: number of loops L, initial learning rate η, learning rate decay factor μ.
Output: adversarially robust student f_L^s.
Load f_0^t from a robust checkpoint
for $l \in \{1, \dots, L\}$ **do**
  $f^s \leftarrow \text{Dropout}(f_{l-1}^s) \triangleleft$ Add dropout to the last layer
  if *l=1* **then**
    $f_l^s \leftarrow \arg\min_{f^s} \mathcal{L}_{IGA}(f^s; \boldsymbol{x}, y, f_l^t) \triangleleft$ Train from scratch with learning rate η according to Algorithm 1
  else
    $f_l^s \leftarrow \arg\min_{f^s} \mathcal{L}_{IGA}(f^s; \boldsymbol{x}, y, f_l^t) \triangleleft$ Fine-tune with learning rate $\mu\eta$ according to Algorithm 1
  $f_{l+1}^t \leftarrow f_l^s$

H. AutoAttack Results

We find that the adversarial robustness obtained by vanilla KDIGA and ISD is only effective to first-order attack like PGD, and it is still vulnerable to AutoAttack. Therefore, we propose to combine an adversarial training step to handle this problem, i.e., we incorporate an one-step FGSM with little overhead into our script to enhance the adversarial robustness against AutoAttack. To be specific, we reuse the input gradient computation in KDIGA for computing the adversarial perturbations to enhance the adversarial robustness against the ensemble attack methods. Results on CIFAR10 can be found in Table 5. We note that there will be a trade-off between the clean accuracy and the robust accuracy once having included adversarial perturbations in the training process. As we only use perturbed images in the training to maximally reduce the overhead, the KDIGA alignment and logits alignment in KD are also conducted on perturbed images instead of the clean images, causing noise in this process. We also tried doing the backward propagation of distillation and input-gradient alignment losses only on the adversarial samples where the teacher predicts correctly. However, no obvious improvement is observed.

The robust accuracy of student models against AutoAttack with different radii and clean accuracy on the ImageNet dataset is shown in Table 6. We note that the attach radii of 0.001 and 0.003 shown in Table 6 are too small for practical robustness evaluation. We show these results for analysis purpose only to illustrate the improvement brought by our proposed method which is obvious with small attack radii. We would leave it a future work to explore better student model and training scheme configurations for large datasets like ImageNet.

I. Combination with ARD

We show two ways to combine our method with adversarial training strategies for KD using ARD [10], i.e., KDIGA-ARD$_C$ and KDIGA-ARD$_A$. The objectives for them are

$$\begin{aligned}\arg\min_{f^s} \mathcal{L}_{IGA_C}(\boldsymbol{x}, y) = \arg\min_{f^s} [&\lambda_{CE}\mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) \\ &+\lambda_{KL}T^2\mathcal{L}_{KL}(f^s(\boldsymbol{x}+\delta)/T, f^t(\boldsymbol{x})/T) \\ &+\lambda_{IGA}\|\nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^t(\boldsymbol{x}), y)\|_2],\end{aligned} \tag{20}$$

$$\begin{aligned}\arg\min_{f^s} \mathcal{L}_{IGA_A}(\boldsymbol{x}, y) = \arg\min_{f^s} [&\lambda_{CE}\mathcal{L}_{CE}(f^s(\boldsymbol{x}), y) \\ &+\lambda_{KL}T^2\mathcal{L}_{KL}(f^s(\boldsymbol{x}+\delta)/T, f^t(\boldsymbol{x}+\delta)/T) + \lambda_{IGA} \\ &\cdot\|\nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^s(\boldsymbol{x}+\delta), y) - \nabla_{\boldsymbol{x}}\mathcal{L}_{CE}(f^t(\boldsymbol{x}+\delta), y)\|_2]\end{aligned} \tag{21}$$

where "IGA_C" is in short for KDIGA-ARD$_C$ and "IGA_A" is in short for KDIGA-ARD$_A$, δ is an adversarial perturbation calculated by solving Eq. 18 as inner maximization. KDIGA-ARD$_C$ is a direct combination of the original ARD formulation

Table 6. Robust accuracy (%) of student models against AutoAttack with different radii and clean accuracy (%) on the ImageNet dataset. Robust accuracy of the teacher models are shown in brackets. The pre-trained student model is denoted with "*" where the distillation is conducted as a fine-tuning process. Other students are all trained from scratch. "ST" means the model is trained following the standard approach without distillation nor adversarial training. "AT" means the model is obtained by adversarial training.

		AutoAttack Attack radius			
Model	**Clean**	**0.001**	**0.003**	**0.005**	**0.01**
ResNet18 (ST)	68.7 (-)	14.3 (-)	0.4 (-)	0.0 (-)	0.0 (-)
ViT-S/16 (ST)	77.6 (-)	48.1 (-)	6.0 (-)	0.5 (-)	0.0 (-)
ViT-S/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	60.0 (77.6)	47.2 (48.1)	25.0 (6.0)	10.1 (0.5)	0.7 (0.0)
ViT-B/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	64.7 (76.3)	49.6 (39.8)	19.4 (5.4)	5.0 (0.6)	0.0 (0.0)
ViT-L/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	65.9 (80.1)	49.6 (46.6)	19.1 (8.5)	5.8 (1.0)	0.0 (0.0)
DEIT-S/16 (ST) $\xrightarrow{\text{KDIGA}}$ ResNet18	63.6 (80.1)	50.0 (0.4)	23.7 (0.0)	7.8 (0.0)	0.1 (0.0)
ResNet50 (AT) $\xrightarrow{\text{KDIGA}}$ ResNet18	54.2 (63.1)	45.9 (47.5)	31.9 (42.5)	19.1 (35.0)	3.9 (30.0)
ResNet50 (AT) $\xrightarrow{\text{KDIGA}}$ ViT-S/16*	77.7 (63.1)	65.3 (47.5)	32.6 (42.5)	13.4 (35.0)	1.1 (30.0)

with our proposed IGA loss on clean samples as an additional regularization. KDIGA-ARD$_A$ further considers perturbed samples in IGA. Their key difference is that KDIGA-ARD$_C$ only aligns student's predictions on perturbed samples with teacher's predictions on clean samples, while KDIGA-ARD$_A$ forces the student to align both predictions and input gradients with the teacher on perturbed samples. We also tried other variants but did not observe notable differences.

J. Limitations

The theoretical proof of the preservation of adversarial robustness relies on the local linearity assumption, which is not necessarily true. And we empirically show that adversarially robust models tend to have better local linearity in Table 3, indicating choosing robust teacher tends to meet such a strong assumption. And we think it is of interest to conduct more analysis on the relationship between adversarial robustness and linearity properties in the future.