Appendix

Table of Contents

A Limitations

We highlight a few key limitations to our results that may be relevant for future work to look at:

1. Our visualizations focus on student-teacher deviations in the top-1 class of the teacher. While this already reveals a systematic pattern across various datasets, this does not capture richer deviations that may occur in the teacher's lower-ranked classes. Examining those would shed light on the "dark knowledge" hidden in the non-target classes.
2. Although we demonstrate the exaggerated bias of Theorem 4.1 in MLPs (Sec D, Fig 20) and CNNs (Sec D, Fig 21), we do not formalize any higher-order effects that may emerge in such multi-layer models. It is possible that the same eigenspace regularization effect propagates down the layers of a network. We show some preliminary evidence in Sec D.7.
3. We do not *exhaustively* characterize when the underlying exaggerated bias of distillation is *(in)sufficient* for improved generalization. One example where this relationship is arguably sufficient is in the case of noise in the one-hot labels (Fig 3). One example where this is insufficient is when the teacher does not fit the one-hot labels perfectly (Fig 3b). A more exhaustive characterization would be practically helpful as it may help us predict when it is worth performing distillation.
4. The effect of the teacher's top-1 accuracy (Sec 5.2) has a further confounding factor which we do not address: the "complexity" of the dataset. For CIFAR-100, the teacher's labels are more helpful than the one-hot labels, even for a mildly-non-interpolating teacher with 4% top-1 error on training data; for CIFAR100, it is only when there is sufficient lack of interpolation that one-hot labels complement the teacher's labels. For the relatively more complex Tiny-Imagenet, the one-hot labels complement teacher's soft labels even when the teacher has 2% top-1 error (Fig 24).

B Proof of Theorem

Below, we provide the proof for Theorem 4.1 that shows that the distilled student converges faster along the top eigendirections than the teacher.

Theorem B.1. *Let $\mathbf{X} \in \mathbb{R}^{n\times p}$ and $\mathbf{y} \in \mathbb{R}^n$ be the p-dimenionsional inputs and labels of a dataset of n examples, where $p > n$. Assume the Gram matrix $\mathbf{X}\mathbf{X}^\top$ is invertible, with n eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ in $\mathbb{R}^p$. Let $\boldsymbol{\beta}(t) \in \mathbb{R}^p$ denote a teacher model at time t, when trained with gradient flow to minimize $\frac{1}{2}\|\mathbf{X}\boldsymbol{\beta}(t) - \mathbf{y}\|^2$, starting from $\boldsymbol{\beta}(0) = \mathbf{0}$. Let $\tilde{\boldsymbol{\beta}}(\tilde{t}) \in \mathbb{R}^p$ be a student model at time $\tilde{t}$, when trained with gradient flow to minimize $\frac{1}{2}\|\mathbf{X}\boldsymbol{\beta}(t) - \mathbf{y}^{te}\|^2$, starting from $\tilde{\boldsymbol{\beta}}(0) = \mathbf{0}$; here $\mathbf{y}^{te} = \mathbf{X}\boldsymbol{\beta}(T^{te})$ is the output of a teacher trained to time $T^{te} > 0$. Let $\beta_k(\cdot)$ and $\tilde{\beta}_k(\cdot)$ respectively denote the component of the teacher and student weights along the k'th eigenvector of the Gram matrix $\mathbf{X}\mathbf{X}^\top$ as:*

$$\beta_k(t) = \boldsymbol{\beta}_k(t) \cdot \mathbf{v}_k, \tag{8}$$

and

$$\tilde{\beta}_k(\tilde{t}) = \tilde{\boldsymbol{\beta}}_k(\tilde{t}) \cdot \mathbf{v}_k. \tag{9}$$

Let $k_1 < k_2$ be two indices for which the eigenvalues satisfy $\lambda_{k_1} > \lambda_{k_2}$, if any exist. Consider any time instants $t > 0$ and $\tilde{t} > 0$ at which both the teacher and the student have converged equally well along the top direction $\mathbf{v}_{k_1}$, in that

$$\beta_{k_1}(t) = \tilde{\beta}_{k_1}(\tilde{t}). \tag{10}$$

Then along the bottom direction, the student has a strictly smaller component than the teacher, as in,

$$\left|\frac{\tilde{\beta}_{k_2}(\tilde{t})}{\beta_{k_2}(t)}\right| < 1. \tag{11}$$

Proof. (of Theorem 4.1)

Recall that the closed form solution for the teacher is given as:

$$\boldsymbol{\beta}(t) = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{A}(t)\mathbf{y} \tag{12}$$

$$\text{where } \mathbf{A}(t) := \mathbf{I} - e^{-t\mathbf{X}\mathbf{X}^\top}. \tag{13}$$

Similarly, by plugging in the teacher's labels into the above equation, the closed form solution for the student can be expressed as:

$$\tilde{\boldsymbol{\beta}}(\tilde{t}) = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\tilde{\mathbf{A}}(\tilde{t})\mathbf{y} \tag{14}$$

$$\text{where } \tilde{\mathbf{A}}(\tilde{t}) := \mathbf{A}(t)\mathbf{A}(T^{\text{te}}). \tag{15}$$

Let $\alpha_k(t), \tilde{\alpha}_k(\tilde{t})$ be the eigenvalues of the k'th eigendirection in $\mathbf{A}(t)$ and $\tilde{\mathbf{A}}(\tilde{t})$ respectively. We are given $\beta_{k_1}(t) = \tilde{\beta}_{k_1}(\tilde{t})$. From the closed form expression for the two models in Eq 12 and Eq 14, we can infer $\alpha_{k_1}(t) = \tilde{\alpha}_{k_1}(\tilde{t})$. Similarly, from the closed form expression, it follows that in order to prove $|\beta_{k_2}(t)| > |\tilde{\beta}_{k_2}(\tilde{t})|$, it suffices to prove $\alpha_{k_2}(t) > \tilde{\alpha}_{k_2}(\tilde{t})$.

For the rest of the discussion, for convenience of notation, we assume $k_1 = 1$ and $k_2 = 2$ without loss of generality. Furthermore, we define $\alpha_1^\star = \alpha_1(t) = \tilde{\alpha}_1(\tilde{t})$.

From the teacher's system of equations in Eq 13, $\alpha_1^\star = 1 - e^{-\lambda_1 t}$. Hence, we can re-write $\alpha_2(t)$ as:

$$\alpha_2(t) = 1 - e^{-\lambda_2 t} \tag{16}$$

$$= 1 - \left(e^{-\lambda_1 t}\right)^{\frac{\lambda_2}{\lambda_1}} \tag{17}$$

$$= 1 - (1 - \alpha_1^\star)^{\frac{\lambda_2}{\lambda_1}}. \tag{18}$$

Similarly for the student, from Eq 15,

$$\alpha_1^\star = (1 - e^{-\lambda_1 \tilde{t}})(1 - e^{-\lambda_1 T^{\text{te}}}). \tag{19}$$

Hence, we can re-write $\tilde{\alpha}_2(\tilde{t})$ as:

$$\tilde{\alpha}_2(\tilde{t}) = (1 - e^{-\lambda_2 \tilde{t}}) \cdot (1 - e^{-\lambda_2 T^{\text{te}}}) \tag{20}$$

$$= \left(1 - \left(e^{-\lambda_1 \tilde{t}}\right)^{\frac{\lambda_2}{\lambda_1}}\right) \cdot \left(1 - \left(e^{-\lambda_1 T^{\text{te}}}\right)^{\frac{\lambda_2}{\lambda_1}}\right) \tag{21}$$

For convenience, let us define $a := e^{-\lambda_1 \tilde{t}}$, $b := e^{-\lambda_1 T^{\text{te}}}$ and $\kappa = \lambda_2/\lambda_1$. Then, rewriting Eq 19, we get

$$\alpha_1^\star = (1 - a)(1 - b). \tag{22}$$

Plugging this into Eq 18,

$$\alpha_2(t) = 1 - (1 - (1 - a)(1 - b))^\kappa. \tag{23}$$

Similarly, rewriting Eq 21, in terms of a, b, κ:

$$\tilde{\alpha}_2(\tilde{t}) = (1 - a^\kappa)(1 - b^\kappa). \tag{24}$$

We are interested in the sign of $\alpha_2(t) - \tilde{\alpha}_2(\tilde{t})$. Let $f(u) = u^\kappa + (a+b-u)^\kappa$. Then, we can write this difference as follows:

$$\begin{aligned}
\alpha_2(t) - \tilde{\alpha}_2(\tilde{t}) &= a^\kappa + b^\kappa - (ab)^\kappa - (1-(1-a)(1-b))^\kappa && (25)\\
&= a^\kappa + b^\kappa - ((ab)^\kappa + (a+b-ab)^\kappa) && (26)\\
&= f(a) - f(a+b(1-a)) = f(b) - f(b+a(1-b)). && (27)
\end{aligned}$$

To prove that last expression in terms of f resolves to a positive value, we make use of the fact that when $\kappa \in (0,1)$, $f(u)$ attains its maximum at $u = \frac{a+b}{2}$, and is monotonically decreasing for $u \in \left[\frac{a+b}{2}, a+b\right]$. Note that κ is indeed in $(0,1)$ because $\lambda_2 < \lambda_1$. Since $\tilde{t} > 0$ and $T^{\text{te}} > 0$, $a \in (0,1)$ and $b \in (0,1)$. Since f is symmetric with respect to a and b, without loss of generality, let a be the larger of $\{a,b\}$.

Since $a < 1$, and $b > 0$, we have $a + b(1-a) > a$. Also since a is the larger of the two, we have $a > \frac{a+b}{2}$. Combining these two, $a + b > a + b(1-a) > a > \frac{a+b}{2}$. Thus, from the monotonic decrease of f for $u \in \left[\frac{a+b}{2}, a+b\right]$, $f(a) > f(a+b(1-a))$. Thus,

$$\alpha_2(t) - \tilde{\alpha}_2(\tilde{t}) > 0, \tag{28}$$

proving our claim.

□

Table 1: Summary of training settings on image data.

Hyperparameter	**CIFAR10* v1**	**CIFAR100 v2**	**Tiny-ImageNet**	**ImageNet**
(based on)		Tian et al. [49]		Cho and Hariharan [7]
Weight decay	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	10^{-4}
Batch size	1024	64	128	1024
Epochs	450	240	200	90
Peak learning rate	1.0	0.05	0.1	0.4
Learning rate warmup epochs	15	1	5	5
Learning rate decay factor	0.1	0.1	0.1	Cosine schedule
Nesterov momentum	0.9	0.9	0.9	0.9
Distillation weight	1.0	1.0	1.0	0.1
Distillation temperature	4.0	4.0	4.0	4.0
Gradual loss switch window	$1k$ steps	$1k$ steps	$10k$ steps	$1k$ steps

C Further experiments on student-teacher deviations

C.1 Details of experimental setup

We present details on relevant hyper-parameters for our experiments.

Model architectures. For all image datasets (CIFAR10, CIFAR100, Tiny-ImageNet, ImageNet), we use ResNet-v2 [15] and MobileNet-v2 [46], models. Specifically, for CIFAR, we consider the CIFAR ResNet-$\{56, 20\}$ family and MobileNet-v2 architectures; for Tiny-ImageNet, we consider the ResNet-$\{50, 18\}$ family and MobileNet-v2 architectures; for ImageNet we consider ResNet-18 family based on the TorchVision implementation. For all ResNet models, we employ standard augmentations as per He et al. [16].

For all text datasets (MNLI, AGNews, QQP, IMDB), we fine-tune a pre-trained RoBERTa [31] model. We consider combinations of cross-architecture- and self-distillation with RoBERTa -Base, -Medium and -Small architectures.

Training settings. We train using minibatch SGD applied to the softmax cross-entropy loss. For all image datasets, we follow the settings in Table 1. For the noisy CIFAR dataset, for 20% of the data we randomly flip the one-hot label to another class. Also note that, we explore two different hyperparameter settings for CIFAR100, for ablation. For all text datasets, we use a batch size of 64, and train for 25000 steps. We use a peak learning rate of 10^{-5}, with 1000 warmup steps, decayed linearly. For the distillation experiments on text data, we use a distillation weight of 1.0. We use temperature $\tau = 2.0$ for MNLI, $\tau = 16.0$ for IMDB, $\tau = 1.0$ for QQP, and $\tau = 1.0$ for AGNews.

For all CIFAR experiments in this section we use GPUs. These experiments take a couple of hours. We run all the other experiments on TPUv3. The ImageNet experiments take around 6-8 hours, TinyImagenet a couple of hours and the RoBERTA-based experiments take $\approx$ 12 hours. Note that for all the later experiments in support of our eigenspace theory (Sec D), we only use a CPU; these finish in few minutes each.

C.2 Scatter plots of probabilities

In this section, we present additional scatter plots of the teacher-student logit-transformed probabilities for the class corresponding to the teacher's top prediction: Fig 7 (for ImageNet), Fig 5,6 (for CIFAR100), Fig 8 (for TinyImagenet), Fig 9 (for CIFAR10), Fig 10 (for MNLI and AGNews settings), Fig 11 (for further self-distillation on QQP, IMDB and AGNews) and Fig 12 (for cross-architecture distillation on language datasets). Below, we qualitatively describe how confidence exaggeration manifests (or does not) in these settings. We attempt a quantitative summary subsequently in Sec C.4.

Image data. First, across *all* the 18 image settings, we observe an underfitting of the low-confidence points on *test* data. Note that this is highly prominent in some settings (e.g., CIFAR100, MobileNet self-distillation in Fig 5 fourth column, second row), but also faint in other settings (e.g., CIFAR100, ResNet56-ResNet20 distillation in Fig 5 second column, second row).

Table 2: Summary of train and test performance of various distillation settings.

Dataset	**Teacher**	**Student**	**Train accuracy**			**Test accuracy**		
			Teacher	**Student (OH)**	**Student (DIST)**	**Teacher**	**Student (OH)**	**Student (DIST)**
CIFAR10	ResNet-56	ResNet-56	100.00	100.00	100.00	93.72	93.72	93.9
	ResNet-56	ResNet-20	100.00	99.95	99.60	93.72	91.83	92.94
	ResNet-56	MobileNet-v2-1.0	100.00	100.00	99.96	93.72	85.11	87.81
	MobileNet-v2-1.0	MobileNet-v2-1.0	100.00	100.00	100.00	85.11	85.11	86.76
CIFAR100	ResNet-56	ResNet-56	99.97	99.97	97.01	72.52	72.52	74.55
	ResNet-56	ResNet-20	99.97	94.31	84.48	72.52	67.52	70.87
	MobileNet-v2-1.0	MobileNet-v2-1.0	99.97	99.97	99.96	54.32	54.32	56.32
	ResNet-56	MobileNet-v2-1.0	99.97	99.97	99.56	72.52	54.32	62.40
(v2 hyperparams.)	ResNet-56	ResNet-56	96.40	96.40	87.61	73.62	73.62	74.40
CIFAR100 (noisy)	ResNet-56	ResNet-56	99.9	99.9	95.6	69.8	69.8	72.7
	ResNet-56	ResNet-20	99.9	91.4	82.8	69.8	64.9	69.2
Tiny-ImageNet	ResNet-50	ResNet-50	98.62	98.62	94.84	66	66	66.44
	ResNet-50	ResNet-18	98.62	93.51	91.09	66	62.78	63.98
	ResNet-50	MobileNet-v2-1.0	98.62	89.34	87.90	66	62.75	63.97
	MobileNet-v2-1.0	MobileNet-v2-1.0	89.34	89.34	82.26	62.75	62.75	63.28
ImageNet	ResNet-18	ResNet-18 (full KD)	78.0	78.0	72.90	69.35	69.35	69.35
	ResNet-18	ResNet-18 (late KD)	78.0	78.0	71.65	69.35	69.35	68.3
	ResNet-18	ResNet-18 (early KD)	78.0	78.0	79.1	69.35	69.35	69.75
MNLI	RoBERTa-Base	RoBERTa-Small	92.9	72.1	72.6	87.4	69.9	70.3
	RoBERTa-Base	RoBERTa-Medium	92.9	88.2	86.8	87.4	83.8	84.1
	RoBERTa-Small	RoBERTa-Small	72.1	72.1	71.0	69.9	69.9	69.9
	RoBERTa-Medium	RoBERTa-Medium	88.2	88.2	85.6	83.8	83.8	83.5
IMDB	RoBERTa-Small	RoBERTa-Small	100.0	100.0	99.1	90.4	90.4	91.0
	RoBERTa-Base	RoBERTa-Small	100.0	100.0	99.9	95.9	90.4	90.5
QQP	RoBERTa-Small	RoBERTa-Small	85.0	85.0	83.2	83.5	83.5	82.5
	RoBERTa-Medium	RoBERTa-Medium	92.3	92.3	90.5	89.7	89.7	89.0
	RoBERTa-Base	RoBERTa-Small	93.5	85.0	85.1	90.5	83.5	84.0
AGNews	RoBERTa-Small	RoBERTa-Small	96.3	96.3	95.7	93.6	93.6	93.3
	RoBERTa-Base	RoBERTa-Medium	99.2	98.4	97.8	95.2	95.2	94.5
	RoBERTa-Base	RoBERTa-Small	99.2	96.3	96.0	95.2	93.6	93.6

Second, on the training data, this occurs in a majority of settings (13 out of 18) except CIFAR100 Mobilenet self-distillation (Fig 5 fourth column) and three of the four CIFAR10 experiments. In all the CIFAR100 settings where this occurs, this is more prominent on training data than on test data.

Third, in a few settings, we also find an overfitting of high-confidence points, indicating a second type of exaggeration. In particular, this occurs for our second hyperparameter setting in CIFAR100 (Fig 6 last column), Tiny-ImageNet with a ResNet student (Fig 8 first and last column).

Language data. In the language datasets, we find the student-teacher deviations to be different in pattern from the image datasets. We find for lower-confidence points, there is typically both significant underfitting and overfitting (i.e., $|Y - X|$ is larger for small X); for high-confidence points, there is less deviation, and if any, the deviation is from overfitting ($Y > X$ for large X). One way to interpret this as the regularization from distillation *deprioritizing* the lower-confidence points.

This behavior is most prominent in four of the settings plotted in Fig 10. We find a weaker manifestation in four other settings in Fig 11. Finally in Fig 12, we report the scenarios where we do not find a meaningful behavior. Nevertheless, there *is* deviation in all the above settings.

Exceptions: In summary, we find patterns in all but the following exceptions:

1. For MobileNet self-distillation on CIFAR100, and for three of the CIFAR10 experiments, we find no underfitting of the lower-confidence points *on the training dataset* (but they hold on test set). Furthermore, in all these four settings, we curiously find an underfitting of the high-confidence points in both test and training data.
2. Our patterns break down in a four of the *cross-architecture* settings of language datasets. This may be because certain cross-architecture effects dominate over the more subtle underfitting effect.

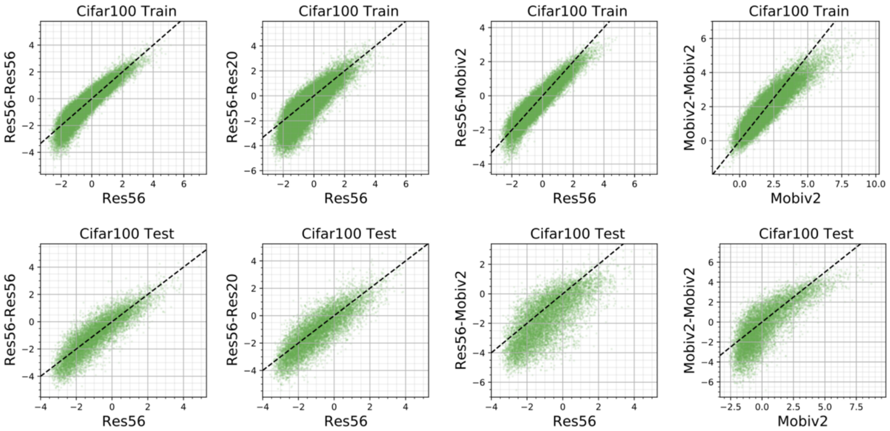


Figure 5: **Teacher-student logit plots for CIFAR100 experiments:** We report plots for various distillation settings involving ResNet56, ResNet20 and MobileNet-v2 (training data on top, test data in the bottom). We find underfitting of the low-confidence points in the training set in all but the MobileNet self-distillation setting. Nevertheless, even in the MobileNet self-distillation setting, we find significant underfitting in the *test* dataset.

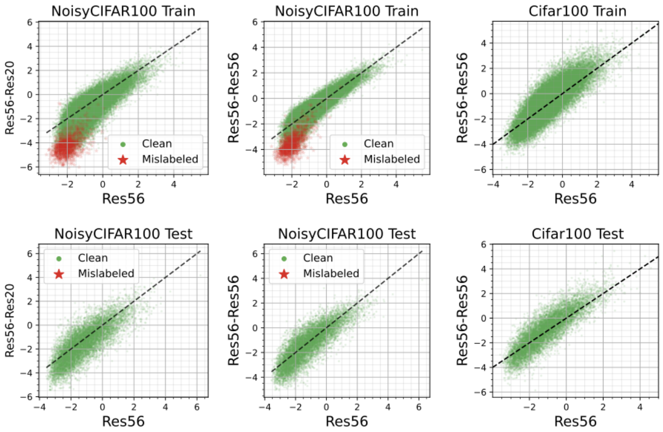
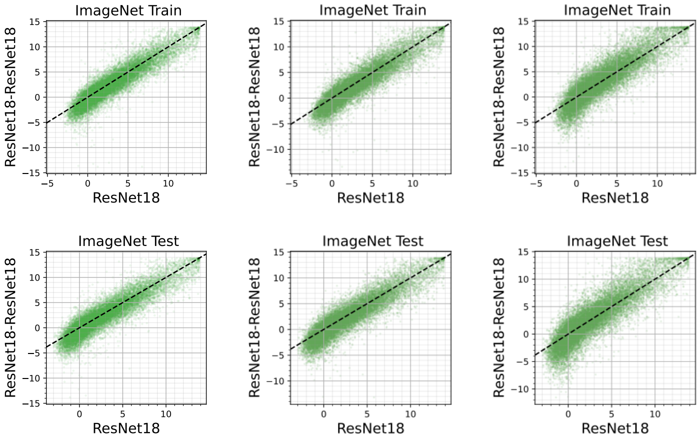


Figure 6: **Teacher-student logit plots for more CIFAR100 experiments:** We report underfitting of low-confidence points for a few other CIFAR100 distillation settings. The first column is self-distillation setting where 20% of one-hot labels are noisy; the second column on the same data, but cross-architecture; the last column is ResNet-56 self-distillation on the original CIFAR100, but with another set of hyperparameters specified in Table 1. Here we also find overfitting of high-confidence points.



(a) Full KD (b) Late-started KD (c) Early-stopped KD

Figure 7: **Teacher-student logit plots for Imagenet experiments:** We conduct Imagenet self-distillation on ResNet18 in three different settings, involving full knowledge distillation, late-started distillation (from exactly mid-way through one-hot training) and early-stopped distillation (again, at the midway point, after which we complete with one-hot training). The plots for the training data are on top, and for test data in the bottom). Note that [7] recommend early-stopped distillation. We find underfitting of low-confidence points in all the settings, with the most underfitting in the last setting.

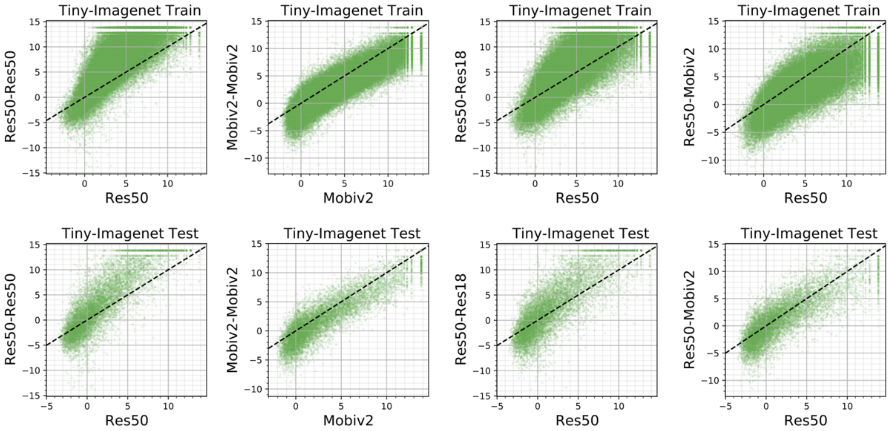


Figure 8: **Teacher-student logit plots for Tiny-Imagenet experiments:** We report plots for various distillation settings involving ResNet50, ResNet18 and MobileNet-v2 (training data on top, test data in the bottom). We find underfitting of the low-confidence points in all the settings. We also find *overfitting of the high-confidence points* when the student is a ResNet.

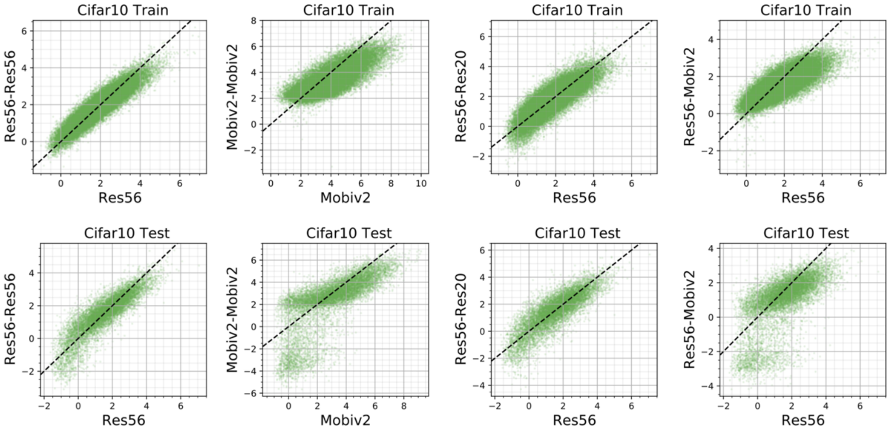
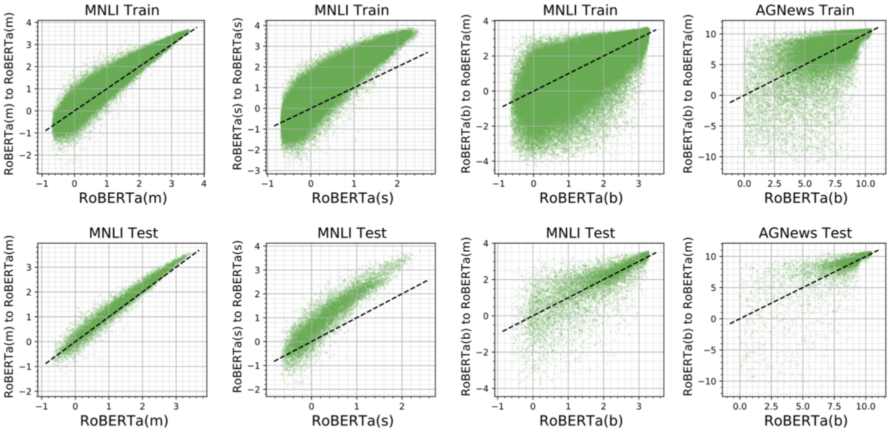


Figure 9: **Teacher-student logit plots for CIFAR10 experiments:** We report plots for various distillation settings involving ResNet56, ResNet20 and MobileNet-v2. We find that the underfitting phenomenon is almost non-existent in the training set (except for ResNet50 to ResNet20 distillation). However the phenomenon is prominent in the test dataset.



(a) Self-distillation in MNLI

(b) Cross-architecture distillation in MNLI and AGNews

Figure 10: **Teacher-student logit plots for MNLI and AGNews experiments:** We report plots for various distillation settings involving RoBERTa models. On the **left**, in the self-distillation settings on MNLI, we find significant underfitting of low-confidence points (and also overfitting), while high-confidence points are significantly overfit. On the **right**, we report cross-architecture (Base to Medium) distillation for MNLI and AGNews. Here, to a lesser extent, we see the same pattern. We interpret this as distillation reducing its "precision" on the lower-confidence points (perhaps by ignoring lower eigenvectors that provide finer precision).

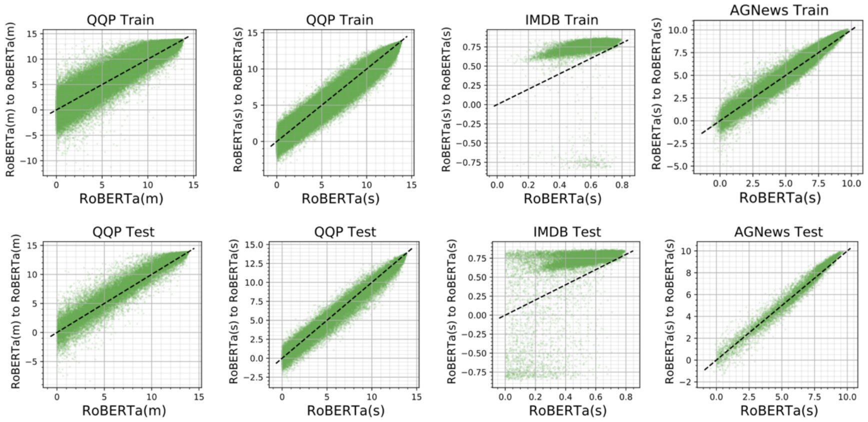


Figure 11: **Teacher-student logit plots for self-distillation in language datasets (QQP, IMDB, AGNews):** We report plots for various self-distillation settings involving RoBERTa models. Except for IMDB training dataset, we find both significant underfitting and overfitting for lower-confidence points (indicating lack of precision), and more precision for high-confidence points. For IMDB test and AGNews, there is an overfitting of the high-confidence points.

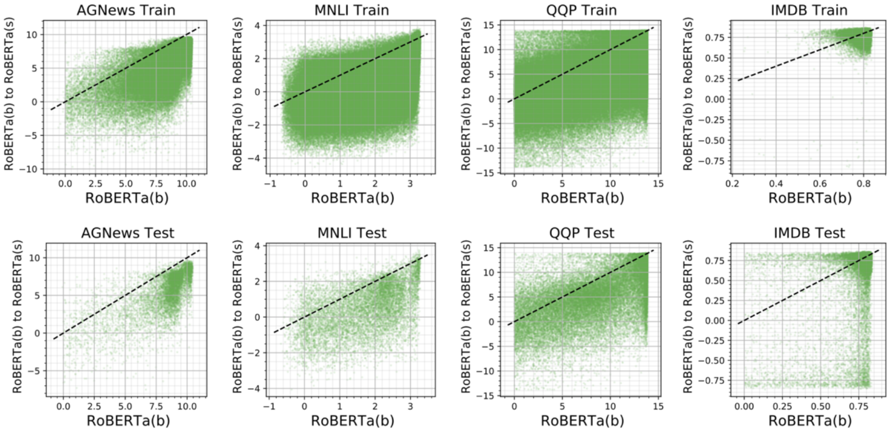


Figure 12: **Teacher-student logit plots for cross-architecture distillation in language datasets (AGNews, QQP, IMDB, MNLI):** We report plots for various cross-architecture distillation settings involving RoBERTa models. While we find significant student-teacher deviations in these settings, our typical patterns do not apply here. We believe that effects due to “cross-architecture gaps” may have likely drowned out the underfitting patterns, which is a more subtle phenomenon that shines in self-distillation settings.

C.3 Teacher's predicted class vs. ground truth class

Recall that in all our scatter plots we have looked at the probabilities of the teacher and the student on the teacher's predicted class i.e., $(p^{\mathsf{te}}_{y^{\mathsf{te}}}(x), p^{\mathsf{st}}_{y^{\mathsf{te}}}(x))$ where $y^{\mathsf{te}} \doteq \operatorname{argmax}_{y' \in [K]} p^{\mathsf{te}}_{y'}(x)$. Another natural alternative would have been to look at the probabilities for the *ground truth class*, $(p^{\mathsf{te}}_{y^\star}(x), p^{\mathsf{st}}_{y^\star}(x))$ where $y^\star$ is the ground truth label. We chose to look at y^{te} however, because we are interested in the "shortcomings" of the distillation procedure where the student only has access to teacher probabilities and not ground truth labels.

Nevertheless, one may still be curious as to what the probabilities for the ground truth class look like. First, we note that the plots look almost identical for the *training dataset* owing to the fact that the teacher model typically fits the data to low training error (we skip these plots to avoid redundancy). However, we find stark differences in the test dataset as shown in Fig 13. In particular, we see that the underfitting phenomenon is no longer prominent, and almost non-existent in many of our settings. This is surprising as this suggests that the student somehow matches the probabilities on the ground truth class of the teacher *despite not knowing what the ground truth class is*.

We note that previous work [33] has examined deviations on ground truth class probabilities albeit in an aggregated sense (at a class-level rather than at a sample-level). While they find that the student tends to have lower ground truth probability than the teacher on problems with label imbalance, they do *not* find any such difference on standard datasets without imbalance. This is in alignment with what we find above.

To further understand the underfit points from Sec C.2 (where we plot the probabilities on teacher's predicted class), in Fig 14, we dissect these plots into four groups: these groups depend on which amongst the teacher and student model classify the point correctly (according to ground truth). We consistently find that the underfit set of points is roughly *the union* of the set of all points where *at least one of the models is incorrect*. This has two noteworthy implications. First, its attempt to deviate from the teacher, the student *corrects* some of the teacher's mistakes. But also, the student introduces *new mistakes* the teacher originally did not make. These may correspond to points which are inherently fuzzy e.g., they are similar to multiple classes.

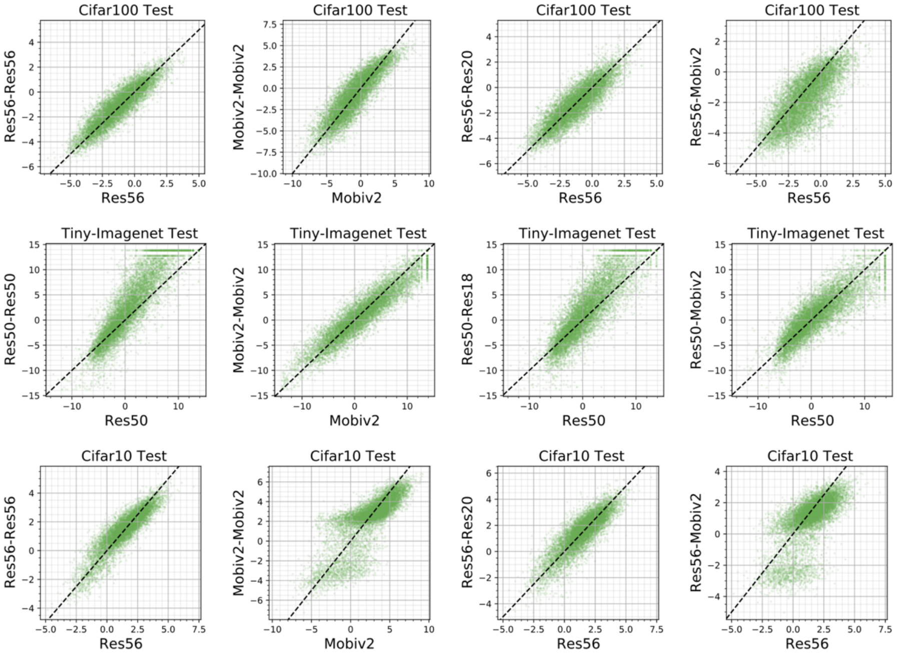
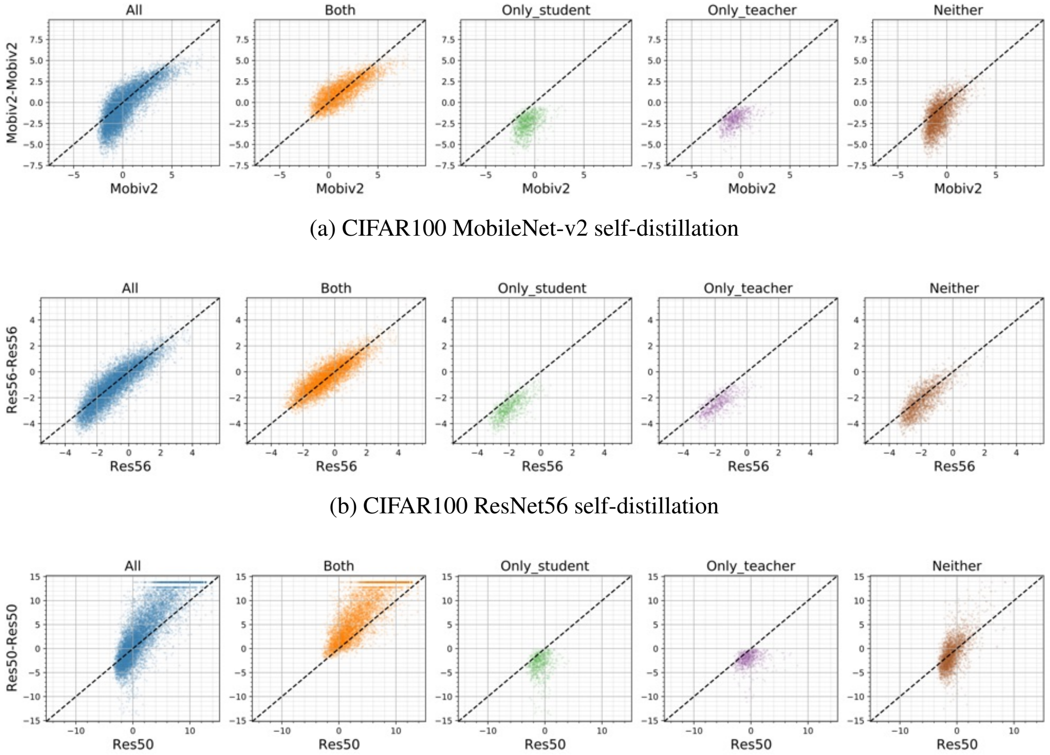


Figure 13: **Scatter plots for ground truth class:** Unlike in other plots where we report the probabilities for the class predicted by the teacher, here we focus on the ground truth class. Recall that the X-axis corresponds to the teacher, the Y-axis to the student, and all the probabilities are log-transformed. Surprisingly, we observe a much more subdued underfitting here, with the phenomenon completely disappearing e.g., in CIFAR100 and CIFAR10 ResNet distillation. This suggests that the student *preserves* the ground-truth probabilities *despite no knowledge of what the ground-truth class is*, while underfitting on the teacher's predicted class.



(a) CIFAR100 MobileNet-v2 self-distillation

(b) CIFAR100 ResNet56 self-distillation

(c) TinyImageNet ResNet50 self-distillation

Figure 14: **Dissecting the underfit points:** Across a few settings on TinyImagenet and CIFAR100, we separate the teacher-student scatter plots of logit-transformed probabilities (for teacher's top predicted class) into four subsets: subsets where both models' top prediction is correct (titled as "Both"), where only the student gets correct ("Only_student"), where only the teacher gets correct ("Only_teacher"), where neither get correct ("Neither"). We consistently find that the student's underfit points are points where at least one of the models go wrong.

Table 3: **Quantification of confidence exaggeration for *self*-distillation settings on *image* datasets:** Slope greater than 1 implies confidence exaggeration. Slope is computed for bottom 25% by teacher's confidence.

Dataset	**Teacher**	**Student**	Slope	
			Train	**Test**
CIFAR10	MobileNet-v2-1.0	MobileNet-v2-1.0	0.22	1.37
	ResNet-56	ResNet-56	0.87	1.13
CIFAR100	MobileNet-v2-1.0	MobileNet-v2-1.0	0.80	1.22
	ResNet-56	ResNet-56	1.26	1.22
(noisy)	ResNet-56	ResNet-56	1.55	1.19
(v2 hyperparameters)	ResNet-56	ResNet-56	1.25	1.31
Tiny-ImageNet	MobileNet-v2-1.0	MobileNet-v2-1.0	1.24	1.22
	ResNet-50	ResNet-50	1.97	1.20
ImageNet	ResNet-18	ResNet-18 (full KD)	1.27	1.22
	ResNet-18	ResNet-18 (late KD)	1.26	1.24
	ResNet-18	ResNet-18 (early KD)	1.38	1.37

Table 4: **Quantification of confidence exaggeration for *cross*-distillation settings on *image* datasets:** Slope greater than 1 implies confidence exaggeration. Slope is computed for bottom 25% by teacher's confidence.

Dataset	**Teacher**	**Student**	Slope	
			Train	**Test**
CIFAR10	ResNet-56	MobileNet-v2-1.0	0.57	1.18
	ResNet-56	ResNet-20	1.05	1.16
CIFAR100	ResNet-56	MobileNet-v2-1.0	0.95	1.03
	ResNet-56	ResNet-20	1.26	1.12
(noisy)	ResNet-56	ResNet-20	1.50	1.60
Tiny-ImageNet	ResNet-50	MobileNet-v2-1.0	1.29	1.08
	ResNet-50	ResNet-18	1.69	1.23

C.4 Quantification of exaggeration

Although we report the exaggeration of confidence levels as a qualitative observation, we attempt a quantification for the sake of completeness. To this end, our idea is to fit a least-squares line $Y = mX + c$ through the scatter plots of $(\phi(p^{\text{te}}_{y^{\text{te}}}(x)), \phi(p^{\text{st}}_{y^{\text{te}}}(x)))$ and examine the slope of the line. If $m > 1$, we infer that there is an exaggeration of confidence values. Note that this is only a proxy measure and may not always fully represent the qualitative phenomenon.

In the image datasets, recall that this phenomenon most robustly occurred in the teacher's low-confidence points. Hence, we report the values of the slope for the bottom 25%-ile points, sorted by the teacher's confidence $\phi(p^{\text{te}}_{y^{\text{te}}}(x))$. Table 3 corresponds to self-distillation and Table 4 to cross-architecture. These values faithfully capture our qualitative observations. In all the image datasets, on test data, the slope *is* greater than 1. The same holds on training data in a majority of our settings, except for the CIFAR-10 settings, and the CIFAR100 settings with a MobileNet student, where we did qualitatively observe the lack of confidence exaggeration.

For the language datasets, recall that there was both an underfitting and overfitting of low-confidence points, but an overfitting of the high-confidence points. To capture this, we report the values of the slope for the top 25%-ile points, Table 5 corresponds to self-distillation and Table 6 to cross-architecture. On test data, the slope is larger than 1 for 7 out of our 12 settings. However, we note that we do not see a perfect agreement between these values and our observations from the plots e.g., in IMDB test data, self-distillation of RoBERTa-small, the phenomenon is strong, but this is not represented in the slope.

Table 5: **Quantification of confidence exaggeration for *self*-distillation settings on *language* datasets:** Slope greater than 1 implies confidence exaggeration. Slope is computed for top 25% points by teacher's confidence.

Dataset	**Teacher**	**Student**	Slope	
			Train	**Test**
MNLI	RoBERTa-Small	RoBerta-Small	1.28	1.30
	RoBERTa-Medium	RoBerta-Medium	0.98	1.00
IMDB	RoBERTa-Small	RoBerta-Small	0.37	0.38
QQP	RoBERTa-Small	RoBerta-Small	1.02	1.01
	RoBERTa-Medium	RoBerta-Medium	0.54	0.59
AGNews	RoBERTa-Small	RoBerta-Small	1.03	1.02

Table 6: **Quantification of confidence exaggeration for *cross*-distillation settings on *language* datasets:** Slope greater than 1 implies confidence exaggeration. Slope is computed for top 25% of points by teacher's confidence.

Dataset	**Teacher**	**Student**	Slope	
			Train	**Test**
MNLI	RoBERTa-Base	RoBerta-Small	1.69	1.68
	RoBERTa-Base	RoBerta-Medium	1.10	1.19
IMDB	RoBERTa-Base	RoBerta-Small	-0.70	0.60
QQP	RoBERTa-Base	RoBerta-Small	23.20	21.53
AGNews	RoBERTa-Base	RoBerta-Small	0.90	1.10
	RoBERTa-Base	RoBerta-Medium	0.88	0.88

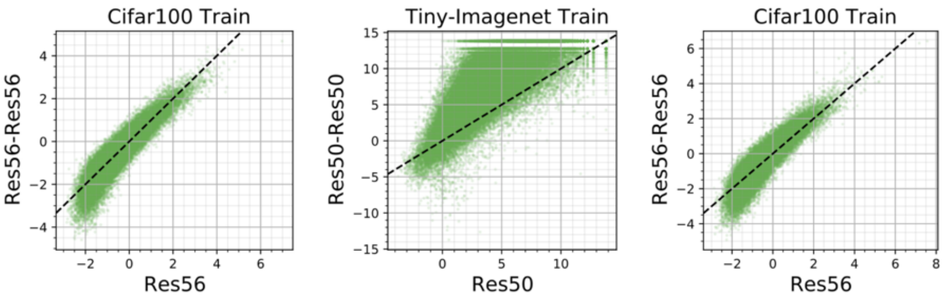


Figure 15: **Underfitting holds for longer runs and for smaller batch sizes:** For the self-distillation setting in CIFAR100 and TinyImagenet **(left two figures)**, we find that the student underfits teacher's low-confidence points even after an extended period of training (roughly $2\times$ longer). On the **right**, we find in the CIFAR100 setting that underfitting occurs even for smaller batch sizes.

C.5 Ablations

We provide some additional ablations in the following section.

Longer training: In Fig 15 (left two images), we conduct experiments where we run knowledge distillation with the ResNet-56 student on CIFAR100 for $2.3\times$ longer ($50k$ steps instead of $21.6k$ steps overall) and with the ResNet-50 student on TinyImagenet for about $2\times$ longer ($300k$ steps over instead of roughly $150k$ steps). We find the resulting plots to continue to have the same underfitting as the earlier plots. It is worth noting that in contrast, in a linear setting, it is reasonable to expect the underfitting to disappear after sufficiently long training. Therefore, the persistent underfitting in the non-linear setting is remarkable and suggests one of two possibilities:

- The underfitting is persistent simply because the student is not trained sufficiently long enough i.e., perhaps, when trained $10\times$ longer, the network might end up fitting the teacher probabilities perfectly.
- The network has reached a local optimum of the knowledge distillation loss and can never fit the teacher precisely. This may suggest an added regularization effect in distillation, besides the eigenspace regularization.

Smaller batch size/learning rate: In Fig 15 (right image), we also verify that in the CIFAR100 setting if we set peak learning rate to 0.1 (rather than 1.0) and batch size to 128 (rather than 1024), our observations still hold. This is in addition to the second hyperparameter setting for CIFAR100 in Fig 6.

A note on distillation weight. For nearly all of our students, we fix the distillation weight to be 1.0 (and so there is no one-hot loss). This is because we are interested in studying deviations under the distillation loss; after all, it is most surprising when the student deviates from the teacher when trained on a pure distillation loss which disincentivizes any deviations.

Nevertheless, for ImageNet, we follow Cho and Hariharan [7] and set the distillation weight to be small, at 0.1 (and correspondingly, the one-hot weight to be 0.9). We still observe confidence exaggeration in this setting in Fig 7. Thus, the phenomenon is robust to this hyperparameter.

Scatter plot for other metrics: So far we have looked at student-teacher deviations via scatter plots of the probabilities on the teacher's top class, *after applying a logit transformation*. It is natural to ask what these plots would look like under other variations. We explore this in Fig 16 for the CIFAR100 ResNet-56 self-distillation setting.

For easy reference, in the top left of Fig 16, we first show the standard logit-transformed probabilities plot where we find the underfitting phenomenon. In the second top figure, we then directly plot the probabilities instead of applying the logit transformation on top of it. We find that the underfitting phenomenon does not prominently stand out here (although visible upon scrutiny, if we examine below the $X = Y$ line for $X \approx 0$). This illegibility is because small probability values tend to concentrate around 0; the logit transform however acts as a magnifying lens onto the behavior of

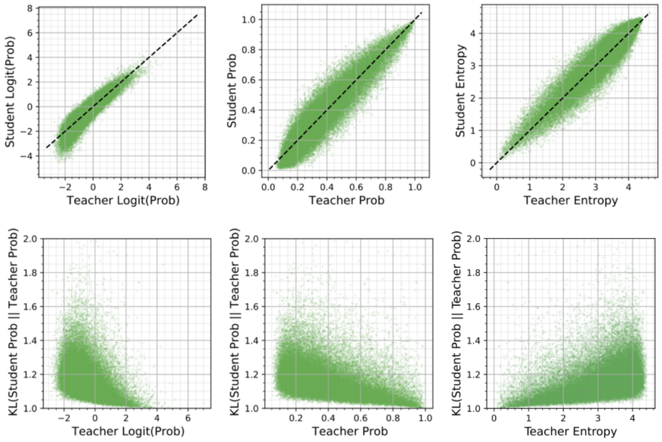


Figure 16: **Scatter plots for various metrics:** While in the main paper we presented scatter plots of logit-transformed probabilities, here we present scatter plots for various metrics, including the probabilities themselves, entropy of the probabilities, and the KL divergence of the student probabilities from the teacher. We find that the KL-divergence plots capture similar intuition as our logit-transformed probability plots. On the other hand, directly plotting the probabilities themselves is not as visually informative.

small probability values. For the third top figure, we provide a scatter plot of entropy values of the teacher and student probability values to determine if the student distinctively deviates in terms of entropy from the teacher. It is not clear what characteristic behavior appears in this plot.

In the bottom plots, on the Y axis we plot the *KL-divergence* of the student's probability from the teacher's probability. Along the X axis we plot the same quantities as in the top row's three plots. Here, across the board, we observe behavior that is aligned with our earlier findings: the KL-divergence of the student tends to be higher on teacher's lower-confidence points, where "lower confidence" can be interpreted as either points where its top probability is low, or points where the teacher is "confused" enough to have high entropy.

Table 7: Summary of the more general training settings used to verify our theoretical claim.

Hyperparameter	**Noisy-MNIST/RandomFeatures**	**MNIST/MLP**	**CIFAR10/CNN**
Width	5000 ReLU Random Features	1000	100
Kernel	-	-	$(6, 6)$
Max pool	-	-	$(2, 2)$
Depth	1	2	3
Number of Classes	10	10	10
Training data size	128	128	8192
Batch size	128	32	128
Epochs	40	20	40
Label Noise	25% (uniform)	None	None
Learning rate	10^{-3}	10^{-4}	10^{-4}
Distillation weight	1.0	1.0	1.0
Distillation temperature	4.0	4.0	4.0
Optimizer	Adam	Adam	Adam

D Further experiments verifying eigenspace regularization

D.1 Description of settings

In this section, we demonstrate the theoretical claims in §4 in practice even in situations where our theoretical assumptions do not hold good. We go beyond our assumptions in the following ways:

1. We consider three architectures: a linear random features model, an MLP and a CNN.
2. All are trained with the cross-entropy loss (instead of the squared error loss).
3. We consider multi-class problems instead of scalar-valued problems.
4. We use a finite learning rate with minibatches and Adam.
5. We test on a noisy-MNIST dataset, MNIST and CIFAR10 dataset.

We provide exact details of these three settings in Table 7.

D.2 Observations

Through the following observations in our setups above, we establish how our insights generalize well beyond our particular theoretical setting:

1. In all these settings, the student fails to match the teacher's probabilities adequately, as seen in Fig 18. This is despite the fact that they both share the same representational capacity. Furthermore, we find that there is a systematic underfitting of the low-confidence points.
2. At the same time, we also observe in Fig 19, Fig 20, Fig 21 that the convergence rate of the student is much faster along the top eigendirections when compared to the teacher in nearly all the pairs of eigendirections that we randomly picked to examine. See §D.3 for how exactly these plots are computed. Note that these plots are shown for the first layer parameters (with respect to the eigenspace of the raw inputs). We show some preliminary evidence that these can be extended to subsequent layers as well (see Fig 22, 23).
3. We also confirm the claim we made in Sec 5.1 to connect the exaggeration of confidence levels to the exaggeration of bias in the eigenspace. In Fig 18 (left), we see that on the mislabeled examples in the NoisyMNIST setting, the teacher has low confidence; the student has even lower confidence on these points. For the sake of completeness, we also show that these noisy examples are indeed fit by the bottom eigendirections in Fig 17. Thus, naturally, a slower convergence along the bottom eigendirections would lead to underfitting of the mislabeled data.

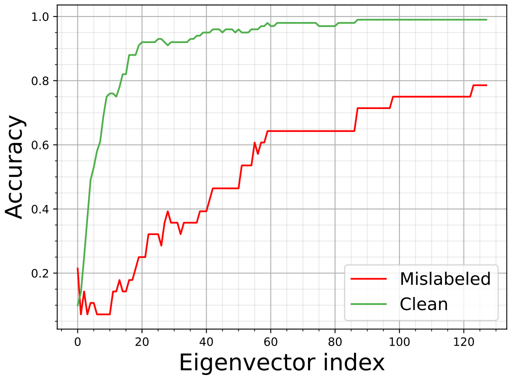


Figure 17: **Bottom eigenvectors help fit mislabeled data:** For the sake of completeness, in the NoisyMNIST setting we report how the accuracy of the model (Y axis) degrades as we retain only components of the weights along the top K eigendirections (K corresponds to X axis). The accuracy on the mislabeled data, as expected, degrades quickly as we lose the bottommost eigenvectors, while the accuracy on clean data is preserved even until K goes as small as 20.

Thus, our insights from the linear regression setting in §4 apply to a wider range of settings. We also find that underfitting happens in these settings, reinforcing the connection between the eigenspace regularization effect and underfitting.

D.3 How eigenvalue trajectories are plotted

How eigendirection trajectories are constructed.

In our theory, we looked at how the component of the weight vector along a data eigendirection would evolve over time. To study this quantity in more general settings, there are two generalizations we must make. First, we have to deal with weight *matrices* or *tensors* rather than vectors. Next, for the hidden weight matrices, it is not clear what corresponding eigenspace we must consider, since its corresponding input is not fixed over time.

Below, we describe how we address these challenges. Our main results in Fig 19, Fig 20, Fig 21 are focused on the first layer weights, where the second challenge is automatically resolved (the eigenspace is fixed to be that of the fixed input data). Later, we show some preliminary extensions to subsequent layers.

How data eigendirections are computed. For the case of the linear model and MLP model, we compute the eigendirections $\mathbf{v}_1, \mathbf{v}_2, \ldots \in \mathbb{R}^d$ directly from the training input features. Here, p is the dimensionality of the (vectorized) data. In the linear model this equals the number of random features, and in the MLP model this is the dimensionality of the raw data (e.g., 784 for MNIST). For the convolutional model, we first take *random* patches of the images of the same shape as the kernel (say (K, K, C) where C is the number of channels). We vectorize these patches into $\mathbb{R}^p$ where $p = K \cdot K \cdot C$ before computing the eigendirections of the data.

How weight components along eigendirections are computed. First we transform our weights into a matrix $\mathbf{W} \in \mathbb{R}^{p \times h}$. For the linear and MLP model, we let $\mathbf{W} \in \mathbb{R}^{p \times h}$ be the weight matrix applied on the p-dimensional data. Here h is the number of outputs of this matrix. In the case of random features, h equals the number of classes, and in the case of the MLP, h is the number of output hidden units of that layer. For the CNN, we flatten the 4-dimensional convolutional weights into $\mathbf{W} \in \mathbb{R}^{p \times h}$ where $p = K \cdot K \cdot C$. Here, h is the number of output hidden units of that layer.

Having appropriately transformed our weights into a matrix $\mathbf{W}$, for any index k, we calculate the component of the weights along that eigendirection as $\mathbf{W}^T \mathbf{v}_k$; we further scalarize this as $\|\mathbf{W}^T \mathbf{v}_k\|_2$. For the plots, we pick two random eigendirections and plot the projection of the weights along those over the course of time.

How to read the plots. In all the plots, the bottom direction is along the Y axis, the top along the X axis. The final weights of either model are indicated by a $\star$. When we say the model shows “implicit

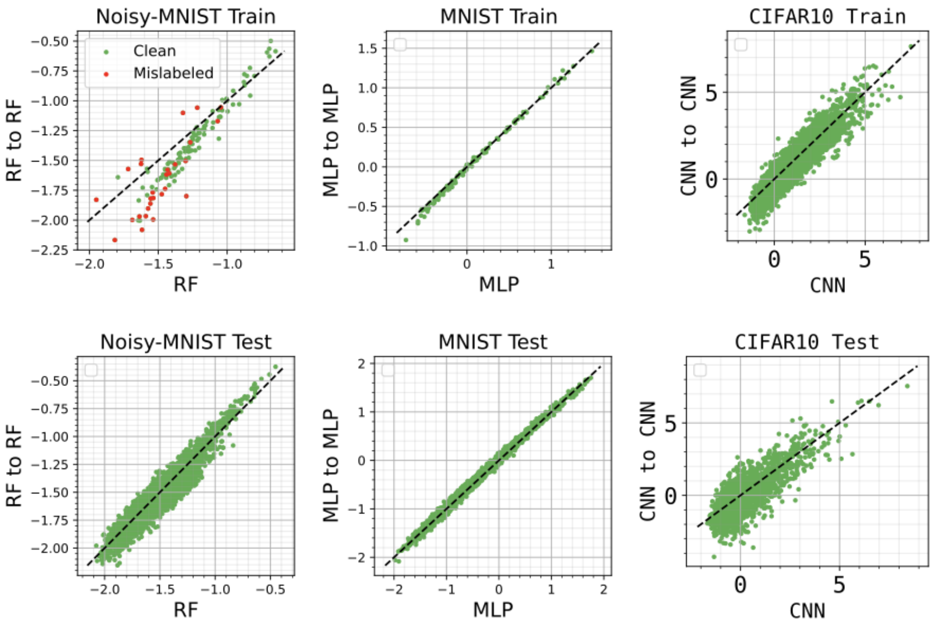


Figure 18: **Confidence exaggeration verifying our theory:** We plot the logit-logit scatter plots, similar to §3, for the three settings in §D — these are also the settings where we verify that distillation exaggerates the implicit bias. Each column corresponds to a different setting, while the top and bottom row correspond to train and test data respectively. Across all the three settings, we find low-confidence underfitting, particularly in the training dataset.

bias", we mean that it converges faster along the top direction in the X axis than the Y axis. This can be inferred by comparing what *fraction* of the X and Y axes have been covered at any point. Typically, we find that the progress along X axis dominates that along the Y axis. Intuitively, when this bias is extreme, the trajectory would reach its final X axis value first with no displacement along the Y axis, and only then take a sharp right-angle turn to progress along the Y axis. In practice, we see a softer form of this bias, where the trajectory takes a "convex" shape, informally put. For the student however, since this bias is strong, the trajectory tends more towards the sharper turn (and is more "strongly convex").

Extending to subsequent layers. The main challenge in extending these plots to a subsequent layer is the fact that these layers act on a time-evolving eigenspace — one that corresponds to the hidden representation of the first layer at any given time. As a preliminary experiment, we fix this eigenspace to be that of the *teacher*'s hidden representation at the *end* of its training. We then train the student with the *same initialization* as that of the teacher so that there is a meaningful mapping between the representation of the two (at least in simple settings, all models originating from the same initialization are known to share interchangeable representations.) Note that we enforce the same initialization in all our previous plots as well. Finally, we plot the student and the teacher's weights projected along the fixed eigenspace of the teacher's representation.

D.4 Verifying eigenspace regularization for random features on NoisyMNIST

Please refer Fig 19.

D.5 Verifying eigenspace regularization for MLP on MNIST

Please refer Fig 20.

D.6 Verifying eigenspace regularization for CNN on CIFAR10

Please refer Fig 21.

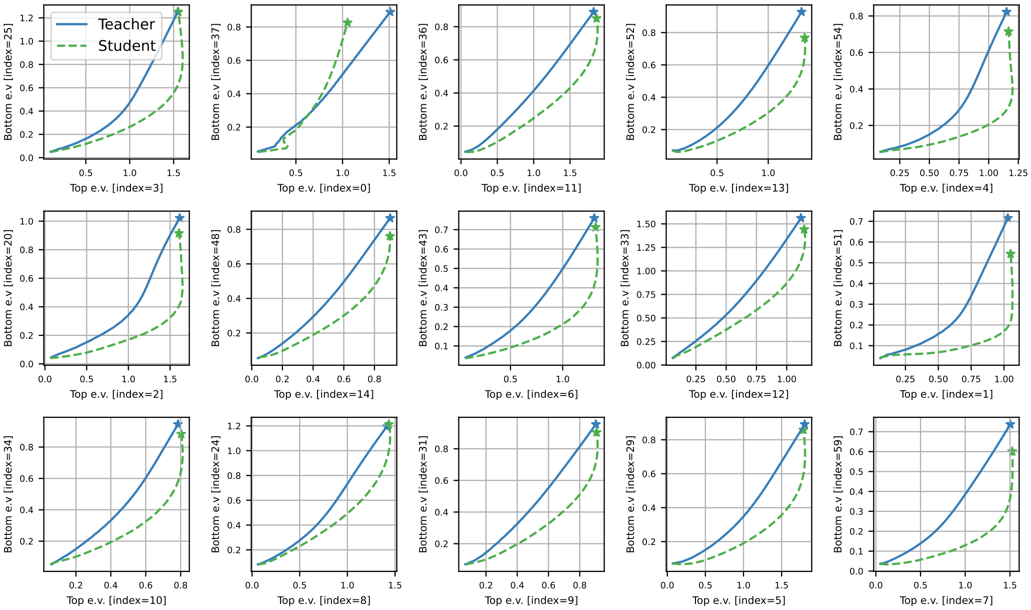


Figure 19: **Eigenspace convergence plots verifying the eigenspace theory for NoisyMNIST-RandomFeatures setting**: In all these plots, the X axis corresponds to the top eigenvector and the Y axis to the bottom eigenvector (see §D for how they are randomly picked). Each plot shows the trajectory projected onto the two eigendirections with the $\star$ corresponding to the final parameters. In all but one case we find that both the student and the teacher converge faster to their final X value, than to their Y value showing that both have a bias towards higher eigendirections. But importantly, this bias is exaggerated for the student in all cases (except the one case in top row, second column), proving our main theoretical claim in §4 in a more general setting with multi-class cross-entropy loss, finite learning rate etc., See §D for discussion.

D.7 Extending to intermediate layers

Please refer Fig 22 and Fig 23.

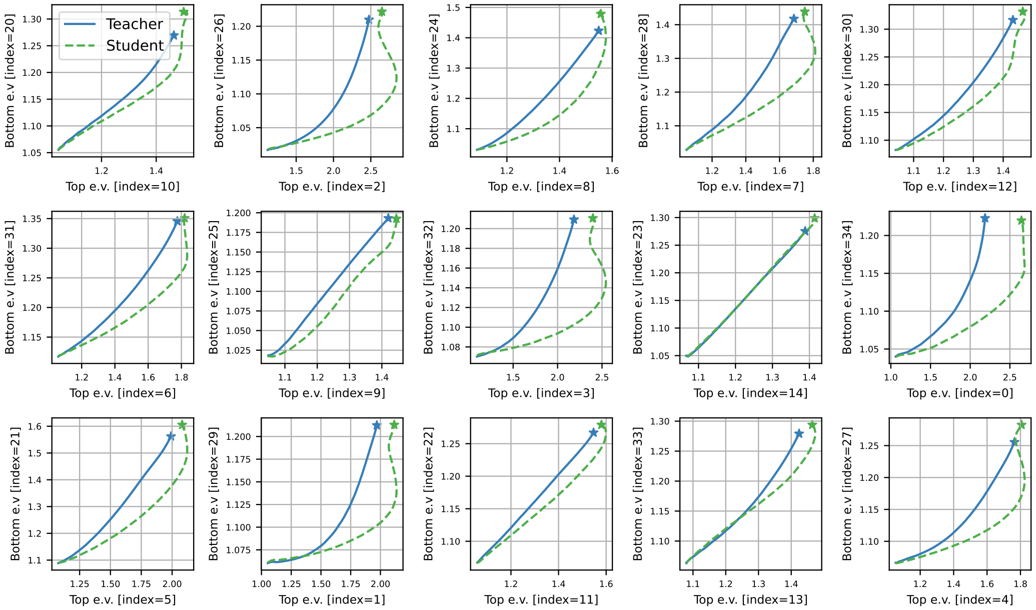


Figure 20: **Eigenspace convergence plots verifying the eigenspace theory for MNIST-MLP setting** : In all cases (except one), we find that the student converges faster to the final X value of the teacher than it does along the Y axis; in the one exceptional case (row 2, col 4), we do not see any difference. This demonstrates our main theoretical claim in §4 in a neural network setting. See §D for discussion.

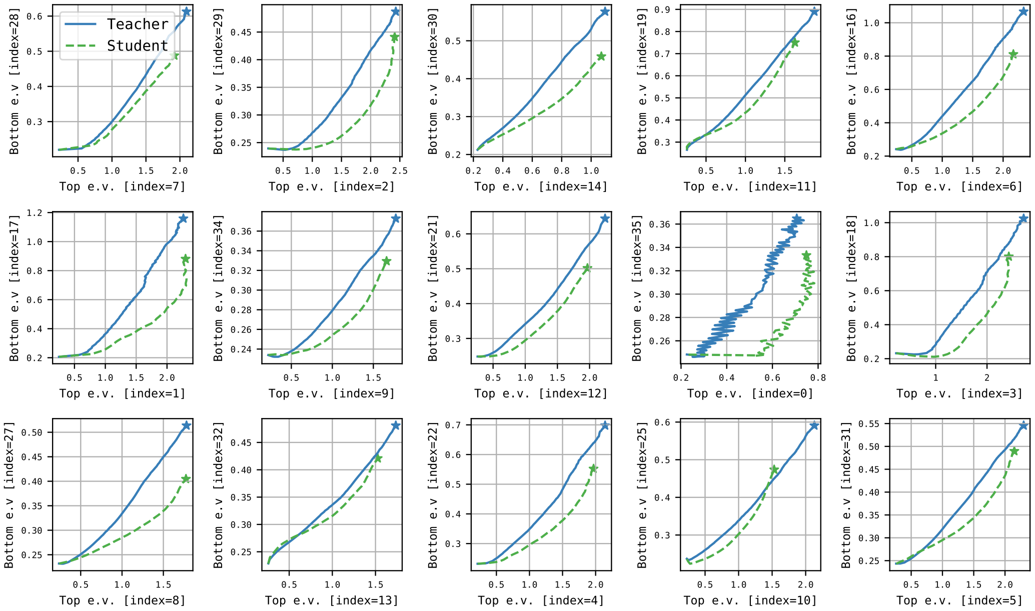


Figure 21: **Eigenspace convergence plots verifying the eigenspace theory for CIFAR10-CNN setting**: In *all* cases, we find that the student converges faster to the final X value of the teacher than it does along the Y axis. This demonstrates our main theoretical claim in §4 in a *convolutional* neural network setting. See §D for discussion.

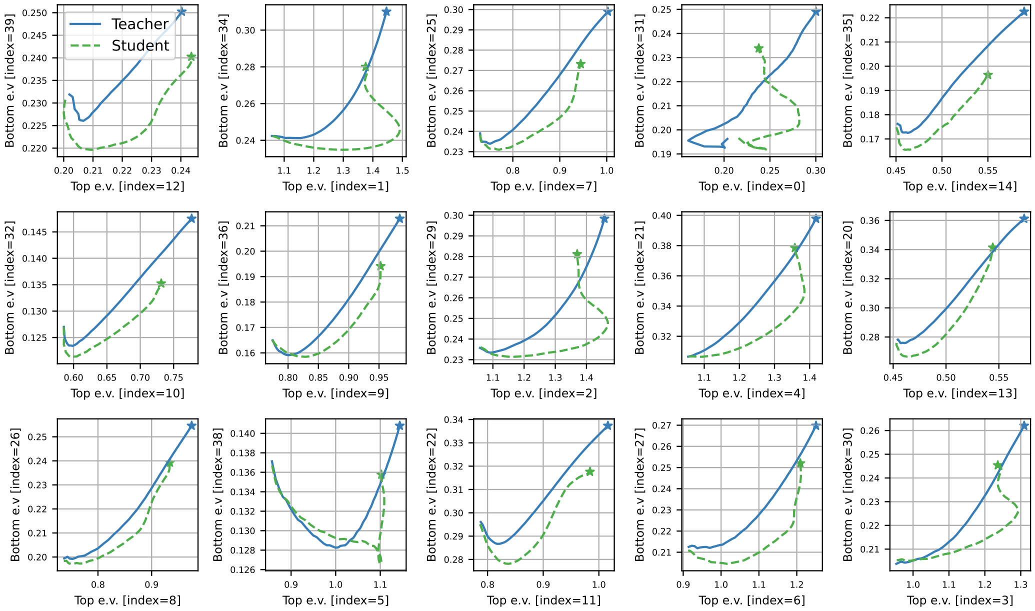


Figure 22: **Eigenspace convergence plots providing preliminary verification the eigenspace theory for the *intermediate* layer in the MNIST-MLP setting**: In all cases (except top row, fourth), we find that the student converges faster to the final X value of the teacher than it does along the Y axis. This demonstrates our main theoretical claim in §4 in an *hidden layer* of a neural network. Note that these plots are, as one would expect, less well-behaved than the first-layer plots in Fig 20. See §D for discussion.

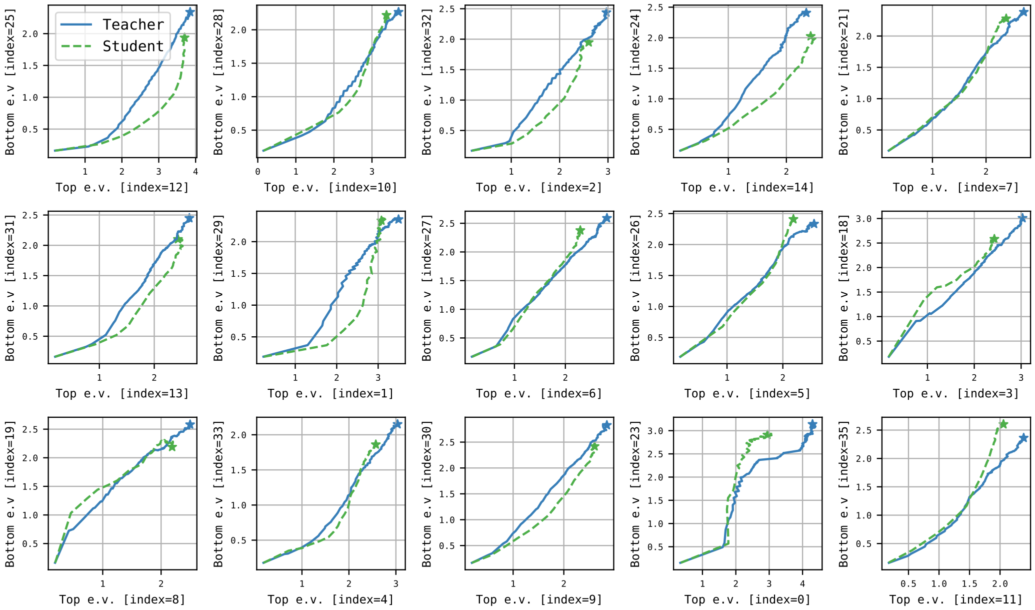


Figure 23: **Eigenspace convergence plots providing preliminary verification of the eigenspace theory for the *intermediate* layer CIFAR-CNN setting**: Here, we find that in a majority of the slices (indexed as 1,2,3,4,6,7,12 and 13 in row-major order), the student has an exaggerated bias than the teacher; in 5 slices (indexed as 2,5,8,9 and 12), there is little change in bias; in 4 slices the student shows a de-exaggerated bias than the teacher. Note that these plots are, as one would expect, less well-behaved than the first-layer plots in Fig 21. See §D for discussion.

E Further experiments on loss-switching

In the main paper, we presented results on loss-switching between one-hot and distillation inspired by prior work [7, 58, 21] that has proposed switching *from* distillation *to* one-hot. We specifically demonstrated the effect of this switch and the reverse, in a controlled CIFAR100 experiment, one with an interpolating and another with a non-interpolating teacher. Here, we present two more results: one with an interpolating CIFAR100 teacher in different hyperparameter settings (see v1 setting in §C.1) and another with a non-interpolating TinyImagenet teacher. These plots are shown in Fig 24. We also present how the logit-logit plots of the student and teacher evolve over time for both settings in Fig 4 and Fig 25.

We make the following observations for the CIFAR100 setting:

1. Corroborating our effect of the interpolating teacher in CIFAR100, we again find that switching to one-hot in the middle of training surprisingly hurts accuracy.
2. Remarkably, we find that for CIFAR100 switching to distillation towards the end of training, is able to regain nearly all of distillation's gains.
3. Fig 25 shows that switching to distillation is able to introduce the confidence exaggeration behavior even from the middle of training; switching to one-hot is able to suppress this behavior.

Note that here training is supposed to end at $21k$ steps, but we have extended it until $30k$ steps to look for any long-term effects of the switch.

In the case of TinyImagenet,

1. For a distilled model, switching to one-hot in the middle of training increases accuracy beyond even the purely distilled model. This is in line with our hypothesis that such a switch would be beneficial under a non-interpolating teacher.
2. Interestingly, for a one-hot-trained model, switching to distillation *is* helpful enough to regain a significant fraction of distillation's gains. However, it does not gain as much accuracy as the distillation-to-onehot switch.
3. Both the one-hot-trained model and the model which switched to one-hot, suffer in accuracy when trained for a long time. This suggests that any switch to one-hot must be done only for a short amount of time.
4. Fig 4 shows that switching to distillation is able to introduce the confidence exaggeration behavior; switching to one-hot is able to suppress this deviation.

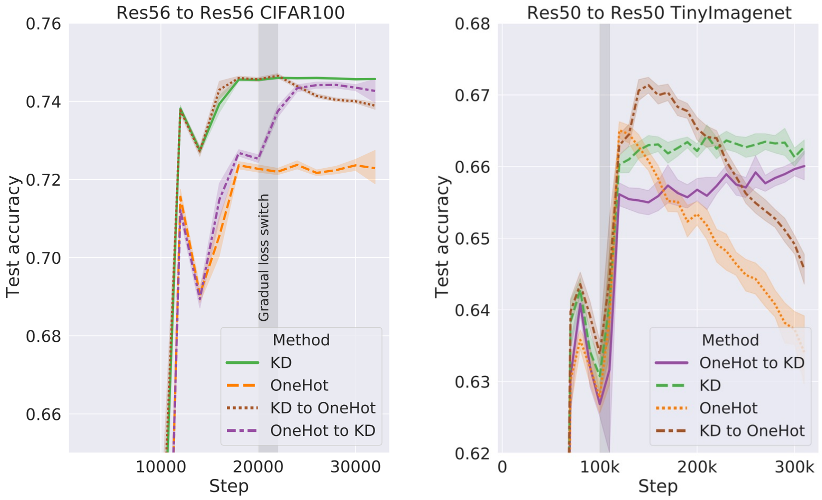
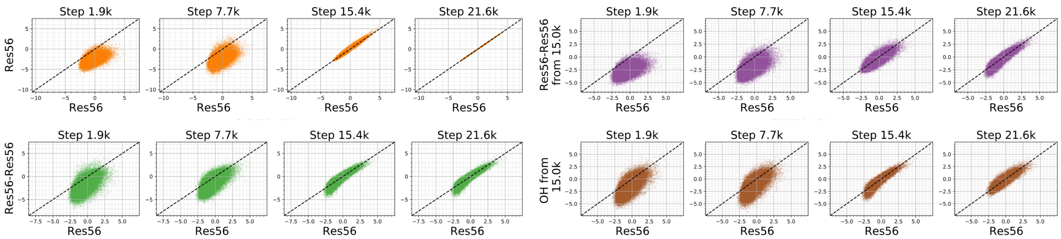


Figure 24: **Trajectory of test accuracy for loss-switching over longer periods of time:** We gradually change the loss for our self-distillation settings in CIFAR100 and TinyImagenet and extend training for a longer period of time. Note that the teacher for the CIFAR100 setting is interpolating while that for the TinyImagenet setting is not. This results in different effects when the student switchs to a one-hot loss, wherein it helps under the non-interpolating teacher and hurts for the interpolating teacher.



(a) One-hot and self-distillation.

(b) Loss-switching to distillation/one-hot at $15k$ steps.

Figure 25: **Evolution of logit-logit plots over various steps of training for CIFAR100 ResNet56 self-distillation setup:** On the **left**, we present plots for one-hot training (**top**) and distillation (**bottom**). On the **right**, we present similar plots the loss switched to distillation (**top**) and one-hot (**bottom**) at $15k$ steps, as discussed in §5.2. From the last two visualized plots in each, observe that switching to distillation introduces (a) underfitting of low-confidence points (b) while switching to one-hot curiously undoes this to an extent.

References

[1] Samira Abnar, Mostafa Dehghani, and Willem H. Zuidema. Transferring inductive biases through knowledge distillation. abs/2006.00555, 2020. URL https://arxiv.org/abs/2006.00555.

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *CoRR*, abs/2012.09816, 2020. URL https://arxiv.org/abs/2012.09816.

[3] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E. Dahl, and Geoffrey E. Hinton. Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations*, 2018.

[4] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, Proceedings of Machine Learning Research. PMLR, 2017.

[5] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10925–10934, June 2022.

[6] Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 535–541, New York, NY, USA, 2006. ACM.

[7] J. H. Cho and B. Hariharan. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4793–4801, 2019.

[8] Andrew Cotter, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, Sashank J. Reddi, and Yichen Zhou. Distilling double descent. *CoRR*, abs/2102.06849, 2021. URL https://arxiv.org/abs/2102.06849.

[9] Wojciech M. Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4278–4287. Curran Associates, Inc., 2017.

[10] Tri Dao, Govinda M Kamath, Vasilis Syrgkanis, and Lester Mackey. Knowledge distillation as semiparametric inference. In *International Conference on Learning Representations*, 2021.

[11] Xiang Deng and Zhongfei Zhang. Can students outperform teachers in knowledge distillation based model compression?, 2021. URL https://openreview.net/forum?id=XZDeL25T12l.

[12] Bin Dong, Jikai Hou, Yiping Lu, and Zhihua Zhang. Distillation $\approx$ early stopping? harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network, 2019.

[13] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 1602–1611, 2018.

[14] Hrayr Harutyunyan, Ankit Singh Rawat, Aditya Krishna Menon, Seungyeon Kim, and Sanjiv Kumar. Supervision complexity and its role in knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8jU7wy7N7mA.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 630–645. Springer, 2016. doi: 10.1007/978-3-319-46493-0_38. URL https://doi.org/10.1007/978-3-319-46493-0_38.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[18] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation. 2020. URL https://arxiv.org/abs/2010.02666.

[19] Fotis Iliopoulos, Vasilis Kontonis, Cenk Baykal, Gaurav Menghani, Khoa Trinh, and Erik Vee. Weighted distillation with unlabeled examples. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS*, 2022.

[20] Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. pages 8580–8589, 2018.

[21] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. Annealing knowledge distillation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2493–2504, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.212. URL https://aclanthology.org/2021.eacl-main.212.

[22] Nandan Kumar Jha, Rajat Saini, and Sparsh Mittal. On the demystification of knowledge distillation: A residual network perspective. 2020.

[23] Guangda Ji and Zhanxing Zhu. Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems*, 2020.

[24] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin L. Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 3491–3501, 2019.

[25] Gal Kaplun, Eran Malach, Preetum Nakkiran, and Shai Shalev-Shwartz. Knowledge distillation: Bad models can be good role models. *CoRR*, 2022.

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[27] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge.

[28] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. pages 8570–8581, 2019.

[29] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*, Proceedings of Machine Learning Research. PMLR, 2020.

[30] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

[31] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, 2019. URL http://arxiv.org/abs/1907.11692.

[32] D. Lopez-Paz, B. Schölkopf, L. Bottou, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, November 2016.

[33] Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. Teacher's pet: understanding and mitigating biases in distillation. *CoRR*, abs/2106.10494, 2021. URL https://arxiv.org/abs/2106.10494.

[34] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 7632–7642. PMLR, 2021.

[35] Hossein Mobahi, Mehrdad Farajtabar, and Peter L. Bartlett. Self-distillation amplifies regularization in hilbert space. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

[36] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems 32*, pages 4696–4705, 2019.

[37] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. What knowledge gets distilled in knowledge distillation? 2022. doi: 10.48550/arXiv.2205.16004. URL https://doi.org/10.48550/arXiv.2205.16004.

[38] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] Minh Pham, Minsu Cho, Ameya Joshi, and Chinmay Hegde. Revisiting self-distillation. *arXiv preprint arXiv:2206.08491*, 2022.

[40] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5142–5151, 2019.

[41] Ilija Radosavovic, Piotr Dollár, Ross B. Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4119–4128, 2018.

[42] Arman Rahbar, Ashkan Panahi, Chiranjib Bhattacharyya, Devdatt Dubhashi, and Morteza Haghir Chehreghani. On the unreasonable effectiveness of knowledge distillation: Analysis in the kernel regime, 2020.

[43] Yi Ren, Shangmin Guo, and Danica J. Sutherland. Better supervisory signals by observing learning paths. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[44] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[47] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, 2021.

[48] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H. Chi, and Sagar Jain. Understanding and improving knowledge distillation. *CoRR*, abs/2002.03532, 2020.

[49] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgpBJrtvS.

[50] Huan Wang, Suhas Lohit, Michael Jeffrey Jones, and Yun Fu. What makes a ”good” data augmentation in knowledge distillation - a statistical perspective. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=6avZnPpk7m9.

[51] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

[52] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification, 2019.

[53] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3902–3910. IEEE, 2020.

[54] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*, 2019.

[55] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[56] Zhilu Zhang and Mert R. Sabuncu. Self-distillation as instance-specific label smoothing. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems*, 2020.

[57] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

[58] Zaida Zhou, Chaoran Zhuge, Xinwei Guan, and Wen Liu. Channel distillation: Channel-wise attention for knowledge distillation. *CoRR*, abs/2006.01683, 2020. URL https://arxiv.org/abs/2006.01683.