Does Knowledge Distillation Really Work?

Samuel Stanton[†] **Pavel Izmailov**[†] **Polina Kirichenko**[†]
Alexander A. Alemi[♣] **Andrew Gordon Wilson**[†]

[†] New York University [♣] Google Research

Abstract

Knowledge distillation is a popular technique for training a small student network to emulate a larger teacher model, such as an ensemble of networks. We show that while knowledge distillation can improve student generalization, it does not typically work as it is commonly understood: there often remains a surprisingly large discrepancy between the predictive distributions of the teacher and the student, even in cases when the student has the capacity to perfectly match the teacher. We identify difficulties in optimization as a key reason for why the student is unable to match the teacher. We also show how the details of the dataset used for distillation play a role in how closely the student matches the teacher — and that more closely matching the teacher paradoxically does not always lead to better student generalization.

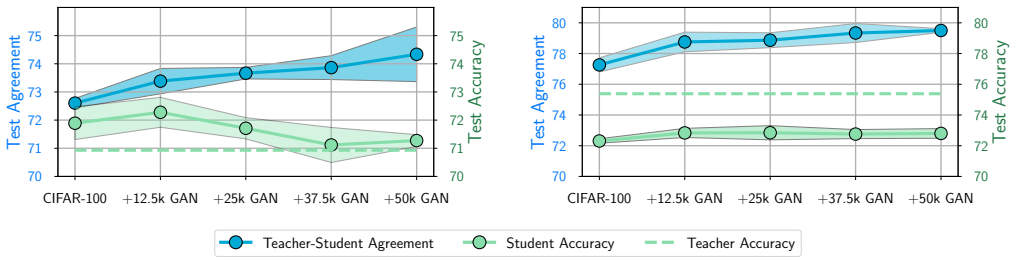
1 Introduction

Large, deep networks can learn representations that generalize well. While smaller, more efficient networks lack the *inductive biases* to find these representations from training data alone, they may have the *capacity* to represent these solutions [e.g., 1, 16, 27, 39]. Influential work on *knowledge distillation* [19] argues that Bucilă et al. [4] "demonstrate convincingly that the knowledge acquired by a large ensemble of models [the teacher] can be transferred to a single small model [the student]". Indeed this quote encapsulates the conventional narrative of knowledge distillation: a student model learns a high-fidelity representation of a larger teacher, enabled by the teacher's soft labels.

Conversely, in Figure 1 we show that with modern architectures knowledge distillation can lead to students with very different predictions from their teachers, even when the student has the capacity to perfectly match the teacher. Indeed, it is becoming well-known that in self-distillation the student fails to match the teacher and, paradoxically, student generalization improves as a result [12, 35]. However, when the teacher is a large model (e.g. a deep ensemble) improvements in fidelity translate into improvements in generalization, as we show in Figure 1(b). For these large models there is still a significant accuracy gap between student and teacher, so fidelity is aligned with generalization.

We will distinguish between *fidelity*, the ability of a student to match a teacher's predictions, and *generalization*, the performance of a student in predicting unseen, in-distribution data. We show that in many cases it is surprisingly difficult to obtain good student fidelity. In Section 5 we investigate the hypothesis that low fidelity is an *identifiability* problem that can be solved by augmenting the distillation dataset. In Section 6 we investigate the hypothesis that low fidelity is an *optimization* problem resulting in a failure of the student to match the teacher even on the original training dataset. We present a summary of our conclusions in Section 7.

Does knowledge distillation really work? In short: *Yes*, in the sense that it often improves student generalization. *No*, in that knowledge distillation often fails to live up to its name, transferring very limited knowledge from teacher to student.



(a) Self-distillation (b) Ensemble distillation

Figure 1: **Evaluating the fidelity of knowledge distillation.** The effect of enlarging the CIFAR-100 distillation dataset with GAN-generated samples. **(a)**: The student and teacher are both single ResNet-56 networks. Student fidelity increases as the dataset grows, but test accuracy decreases. **(b)**: The student is a single ResNet-56 network and the teacher is a 3-component ensemble. Student fidelity again increases as the dataset grows, but test accuracy now slightly increases. The shaded region corresponds to $\mu \pm \sigma$, estimated over 3 trials.

2 Related Work

Knowledge distillation can improve model efficiency [33, 39], unsupervised domain adaptation [32], improved object detection [8], model transparency [42], and adversarial robustness [13, 37].

Seminal work by Bucilă et al. [4] showed that teacher-ensembles with thousands of simple components could be compressed into a single shallow network that matched or outperformed its teacher. Other early work proposed distilling ensembles of shallow networks into a single network [48], an idea which resonates with more recent work on the distillation of deep ensembles [1, 6, 40, 44, 46]. Recently Fakoor et al. [11] developed a data-augmentation scheme for the distillation of large ensembles of simple models for tabular data, achieving impressive results on a wide range of tabular benchmarks. Malinin et al. [29] proposed a method to model the implicit distribution over predictive distributions from which the ensemble component predictive distributions are drawn, rather than just the ensemble model average.

Our work focuses explicitly on student fidelity, decoupling our understanding of good fidelity from good generalization. We show that achieving good fidelity is extremely difficult, even with a variety of interventions, and seek to *understand*, by systematically considering several hypotheses, why knowledge distillation does not produce high fidelity students for modern architectures and datasets. In contrast, the distillation literature focuses largely on improving student generalization, without particularly distinguishing between fidelity and generalization.

For example, concurrent work by Beyer et al. [3] also observes that high student fidelity is both desirable and difficult to achieve, but focuses most heavily on practical modifications to the distillation procedure for the best student top-1 classification accuracy. In this paper we investigate many of the same prescriptions, including careful treatment of data augmentation (such as showing the teacher and student the exact same points), the addition of MixUp, and extended training duration. We also find that such interventions do improve student accuracy, but there still remains a large discrepancy between the predictive distributions of the teacher and the student. We also investigate multiple optimizers. While we do not pursue Shampoo [15] specifically, Beyer et al. [3] find similar qualitative results for Shampoo and Adam, besides faster convergence for Shampoo. As is common in the knowledge distillation literature, Beyer et al. [3] does not make a particular distinction between fidelity and generalization.

3 Preliminaries

We will focus on the supervised classification setting, with input space $\mathcal{X}$ and label space $\mathcal{Y}$, where $|\mathcal{Y}| = c$. Let $f : \mathcal{X} \times \Theta \to \mathbb{R}^c$ be a classifier parameterized by $\theta \in \Theta$ whose outputs define a categorical predictive distribution over $\mathcal{Y}$, $\hat{p}(y = i|\mathbf{x}) = \sigma_i(f(\mathbf{x}, \theta))$, where $\sigma_i(\mathbf{z}) := \exp(z_i)/\sum_j \exp(z_j)$ is the softmax link function. We will often refer to the outputs of a classifier $\mathbf{z} := f(\mathbf{x}, \theta)$ as *logits*. For convenience, we will use t and s as shorthand for f_{teacher} and f_{student}, respectively. When the teacher is an m-component ensemble, the component logits $(\mathbf{z}_1, \ldots, \mathbf{z}_m)$, where $\mathbf{z}_i = f_i(\mathbf{x}, \theta_i)$, are combined to form the teacher logits: $\mathbf{z}_t = \log\left(\sum_{i=1}^m \sigma(\mathbf{z}_i)/m\right)$. These combined logits correspond to the predictive distribution of the ensemble model average.

3.1 Knowledge Distillation

Hinton et al. [19] proposed a simple approach to knowledge distillation. The student minimizes a weighted combination of two objectives, $\mathcal{L}_s := \alpha\mathcal{L}_{\text{NLL}} + (1-\alpha)\mathcal{L}_{\text{KD}}$, where $\alpha \in [0, 1)$. Specifically,

$$\mathcal{L}_{\text{NLL}}(\mathbf{z}_s, \mathbf{y}) := -\sum_{j=1}^{c} y_j \log \sigma_j(\mathbf{z}_s), \quad \mathcal{L}_{\text{KD}}(\mathbf{z}_s, \mathbf{z}_t) := -\tau^2 \sum_{j=1}^{c} \sigma_j\left(\frac{\mathbf{z}_t}{\tau}\right) \log \sigma_j\left(\frac{\mathbf{z}_s}{\tau}\right). \tag{1}$$

$\mathcal{L}_{\text{NLL}}$ is the usual supervised cross-entropy between the student logits $\mathbf{z}_s$ and the one-hot labels $\mathbf{y}$. Recalling that $\text{KL}(p||q) = \sum_j p_j(\log q_j - \log p_j)$, we see that $\mathcal{L}_{\text{NLL}}$ is equivalent (up to a constant) to the KL from the empirical data distribution to the student predictive distribution ($\hat{p}_s$). $\mathcal{L}_{\text{KD}}$ is the added knowledge distillation term that encourages the student to match the teacher. It is the cross-entropy between the teacher and student predictive distributions $\hat{p}_t = \sigma(\mathbf{z}_t)$ and $\hat{p}_s = \sigma(\mathbf{z}_s)$, *both* scaled by a temperature hyperparameter $\tau > 0$. If $\tau = 1$ then $\mathcal{L}_{\text{KD}}$ is similarly equivalent to the KL from the teacher to the student, $\text{KL}(\hat{p}_t||\hat{p}_s)$. Since we focus on distillation fidelity, we choose $\alpha = 0$ for all experiments in the main text to avoid any confounding from true labels, but we also include a limited ablation of α in Figure 14 in Appendix C.2 for the curious reader.

As $\tau \to +\infty$, $\nabla_{\mathbf{z}_s}\mathcal{L}_{\text{KD}}(\mathbf{z}_s, \mathbf{z}_t) \approx \mathbf{z}_t - \mathbf{z}_s$, and thus in the limit $\nabla_{\mathbf{z}_s}\mathcal{L}_{\text{KD}}$ is approximately equivalent to $\nabla_{\mathbf{z}_s}||\mathbf{z}_t - \mathbf{z}_s||_2^2/2$, assigning equal significance to every class logit, regardless of its contribution to the predictive distribution. In other words τ determines the "softness" of the teacher labels, which in turn determines the allocation of student capacity. If the student is much smaller than the teacher, the student capacity can be focused on matching the teacher's top-k predictions, rather than matching the full teacher distribution by choosing a moderate value (e.g. $\tau = 4$).

The teacher and student often share at least some training data. It is also common to enlarge the student training data in some way (e.g. incorporating unlabeled examples as in Ba and Caruana [1]). When there is a possibility of confusion, we will refer to the student's training data as the *distillation data* to distinguish it from the teacher's training data.

3.2 Metrics and Evaluation

To measure generalization, we report top-1 accuracy, negative log-likelihood (NLL) and expected calibration error (ECE) [14]. To measure fidelity, we report the following:

$$\text{Average Top-1 Agreement} := \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\operatorname*{argmax}_j \sigma_j(\mathbf{z}_{t,i}) = \operatorname*{argmax}_j \sigma_j(\mathbf{z}_{s,i})\}, \tag{2}$$

$$\text{Average Predictive KL} := \frac{1}{n}\sum_{i=1}^{n} \text{KL}\left(\hat{p}_t(\mathbf{y}|\mathbf{x}_i) \;||\; \hat{p}_s(\mathbf{y}|\mathbf{x}_i)\right), \tag{3}$$

Eqn. (2) is the average *agreement* between the student and teacher's top-1 label. Eqn. (3) is the average KL divergence from the predictive distribution of the teacher to that of the student, a measure of fidelity sensitive to all of the labels.

While improvements in generalization metrics are relatively easy to understand, interpreting fidelity metrics requires some care. For example, suppose we have three independent models: f_1, f_2, and f_3 that respectively achieve 55%, 75%, and 95% test accuracy. f_1 and f_3 can agree on at most 60% of points, whereas f_2 and f_3 agree on at least 70%, but it would obviously be incorrect to make any claim about f_2 being a better distillation of f_3 since each model was trained completely independently. To account for such confounding when evaluating the distillation of a student s from a teacher t, we also evaluate another student s' distilled through an identical procedure from an independent teacher. By comparing the fidelity of (t, s) and (t, s') we can distinguish between a generic improvement in generalization and an improvement specifically to fidelity. If s and s' have comparable fidelity, then the students agree with the teacher at many points because they generalize well, and not the reverse.

4 Knowledge Distillation Transfers Knowledge Poorly

In this section, we present evidence that we are not able to distill large networks such as a ResNet-56 with high fidelity, and discuss why high fidelity an important objective.

4.1 When is knowledge transfer successful?

We first consider the easy task of distilling a LeNet-5 teacher into an identical student network as a motivating example. We train the teacher on a random subset of 200 examples from the MNIST training set for 100 epochs, resulting in a 84% to 86% teacher test accuracy across different subsets.[1] We then distill the teacher using the full MNIST train dataset with 60,000 examples, as well as 25%, 50%, and 100% of the EMNIST train dataset [10]. The EMNIST train set contains 697,932 images.

In Figure 2 we see that knowledge distillation works as expected. With enough examples the student learns to make the same predictions as the teacher (over 99% top-1 test agreement). Notably, in this case, self-distillation does not *improve* generalization, since the slight difference between the teacher and student accuracy is explained by variance between trials.

Now we consider a more challenging task: distilling a ResNet-56 teacher trained on CIFAR-100 into an identical student network (Figure 1, left). Since no dataset drawn from the same distribution as CIFAR-100 is publicly available, to augment the distillation data, we instead combined samples from an SN-GAN [34] pre-trained on CIFAR-100 with the original CIFAR-100 train dataset. Appendix A.3 details the hyperparameters and training procedure for the GAN, teacher, and student.

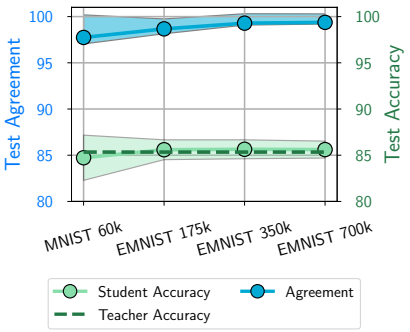


Figure 2: LeNet-5 self-distillation on MNIST with additional distillation data. The shaded region corresponds to $\mu \pm \sigma$, estimated over 3 trials.

Like the MNIST experiment, as we enlarge the distillation dataset the student fidelity improves. However, in this case the improvement is modest, with the fidelity reaching nowhere near 99% test agreement. Since a ResNet-56 has many more parameters than a LeNet-5, it is possible that the student simply has not seen enough examples to perfectly emulate the teacher, a hypothesis we discuss in more detail in Section 5.1. Also, like the MNIST experiment, as the distillation dataset grows the student accuracy approaches the teacher's. *Unlike* the MNIST experiment, the student test accuracy is higher than the teacher's when the distillation dataset is small, so increasing fidelity *decreases* student generalization.

[1] We took only a subset of the MNIST train set since otherwise every teacher network as well as the ensemble would achieve over 99% test accuracy.

4.2 What can self-distillation tell us about knowledge distillation in general?

We have seen in Figure 1(a) that with self-distillation the student can exceed the teacher performance, in accordance with Furlanello et al. [12]. This result is only possible by virtue of failing at the distillation procedure: if the student matched the teacher perfectly then the student could not outperform the teacher. On the other hand, if the teacher generalizes significantly better than an independently trained student, we would expect the benefits of fidelity to dominate other regularization effects associated with not matching the teacher. This setting reflects the original motivation for knowledge distillation, where we wish to faithfully transfer the representation discovered by a large model or ensemble of models into a more efficient student.

In Figure 1(b) we see that if we move from self-distillation to the distillation of a 3 ResNet-56 teacher ensemble, fidelity becomes positively correlated with generalization. But there is still a significant gap in fidelity, even after the distillation set is enlarged with $50k$ GAN samples. In practice, the gap remains large enough that higher fidelity students do not always have better generalization, and the regularization effects we see in self-distillation do play a role for more broadly understanding student generalization. We will indeed show in Section 5 that higher fidelity students do not always generalize better, even if the teacher generalizes much better than the student.

4.3 If distillation improves generalization, why care about fidelity?

While knowledge distillation does often improve generalization, understanding the relationship between fidelity and generalization, and how to maximize fidelity, is important for several reasons.

Distilling large teacher models. Knowledge distillation was initially motivated as a means to deploy powerful models to small devices or low-latency controllers [e.g., 9, 18, 23, 45, 47]. While in self-distillation generalization and fidelity are in tension, there is often a significant disparity in generalization between large teacher models and smaller students. We have seen this disparity in Figure 1(b). We additionally show in Figure 10 in Appendix B.2 that as we increase the number of ensemble components, the generalization disparity between teacher and distilled student increases. Improving student fidelity is the most obvious way to close the generalization disparity between student and teacher in these settings.

Interpretability and reliability. Knowledge distillation has been identified as a means to *transfer representations* discovered by large black-box models into simpler more interpretable models, for example to provide insights into medical diagnostics, or discovering rules for understanding sentiment in text [e.g., 20, 21, 5, 28, 7]. The ability to perform this transfer could have extraordinary scientific consequences: large models can often discover structure in data that we would not have anticipated a priori. Moreover, we often want to transfer properties such as well-calibrated uncertainties or robustness, which have been well-established for larger models, so that we can safely deploy more efficient models in their place [31]. In both cases, achieving good distillation fidelity is crucial.

Understanding. The name *knowledge distillation* implies we are transferring knowledge from the teacher to the student. For this reason, improved student generalization as a consequence of a distillation procedure is sometimes conflated with fidelity. Decoupling fidelity and generalization, and explicitly studying fidelity, is foundational to understanding how knowledge distillation works and how we can make it more useful across a variety of applications.

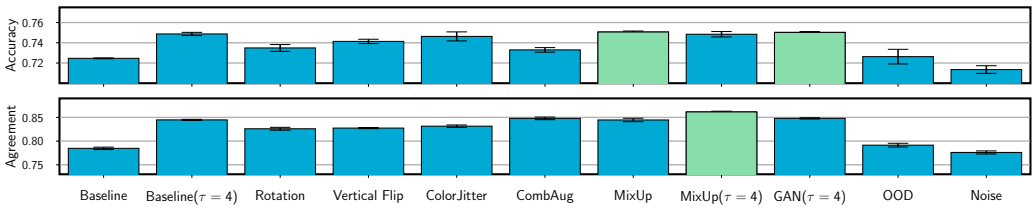


Figure 3: **Data augmentation and distillation**: Test accuracy and teacher-student agreement when distilling a 5-component ResNet-56 teacher ensemble into a ResNet-56 student on CIFAR-100 with varying augmentation policies. The best performing policy is shown in green, results averaged over 3 runs. Additional metrics are reported in Figure 12 in Appendix C. Mixup and GAN augmentation provide the best generalization, and Mixup($\tau = 4$) provides the best fidelity. The baseline policy (crops and flips) with $\tau = 4$ is a surprisingly strong baseline. The error bars indicate $\pm\sigma$.

4.4 Possible causes of low distillation fidelity

If we are able to match the student model to the teacher on a comprehensive distillation dataset, we expect it to match on the test data as well, achieving high distillation fidelity[2]. Possible causes of the poor distillation fidelity in our CIFAR-100 experiments include:

Architecture – Low fidelity could be specific to ResNet-like architectures, an explanation we rule out by showing similar results with VGG networks [41] in Figure 8 in Appendix B.1.

Student Capacity – We observe low fidelity even in the self-distillation setting, so we can rule out student capacity as a primary cause, but we also confirm in Figure 13 in Appendix C.1 that increasing the student capacity has almost no effect on fidelity in the ensemble-distillation setting.

Identifiability (Section 5) – the distillation data is insufficient to distinguish high-fidelity and low-fidelity students. In other words, matching the teacher predictions on the distillation dataset does not lead to matching predictions on the test data.

Optimization (Section 6) – we are unable to solve the distillation optimization problem sufficiently well. The student does not agree with the teacher on test because it does not even agree on train.

5 Identifiability: Are We Using the Right Distillation Dataset?

We investigate whether it is possible to attain the level of fidelity observed with LeNet-5s on MNIST with ResNets on CIFAR-100 by addressing the *identifiability* problem — have we shown the student enough of the right input-teacher label pairs to define the solution we want?

5.1 Should we do more data augmentation?

Data augmentation is a simple and practical method to increase the support of the distillation data distribution. If identifiability is a primary cause of poor distillation fidelity, using a more extensive data augmentation strategy during distillation should improve fidelity.

To test this hypothesis, we evaluated the effect of several augmentation strategies on student fidelity and generalization. In Figure 3, the teacher is a 5-component ensemble of ResNet-56 networks trained on CIFAR-100 with the *Baseline* augmentation strategy: horizontal flips

[2]See, for example, Lemma 1 in Fakoor et al. [11].

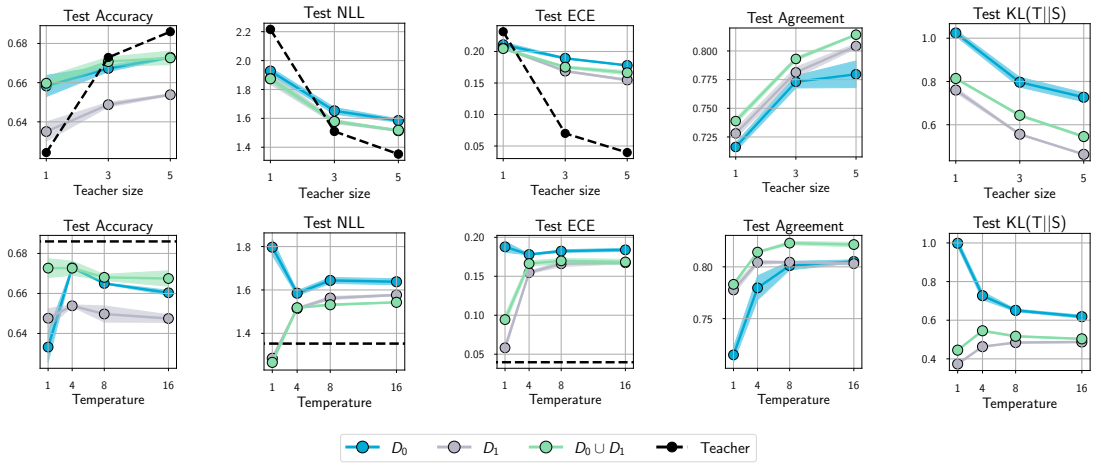


Figure 4: **Data recycling and distillation**: results on subsampled CIFAR-100. **Top:** We fix the temperature ($\tau = 4$) and vary the number of ensemble components (m), comparing students distilled on the same dataset as the teacher ($\mathcal{D}_0/\mathcal{D}_0$), a reserved dataset ($\mathcal{D}_0/\mathcal{D}_1$), or both ($\mathcal{D}_0/\mathcal{D}_0 \cup \mathcal{D}_1$). Distilling on both produces the best result, while distilling on $\mathcal{D}_0$ increases accuracy and decreases fidelity, relative to $\mathcal{D}_1$. **Bottom:** We repeat the experiment, but fix $m = 3$ and vary τ. The shaded region corresponds to $\mu \pm \sigma$, estimated over 3 trials.

and random crops. We report the student accuracy and teacher-student agreement for each augmentation strategy, and also include results for *Baseline* with $\tau = 1$ and $\tau = 4$ to demonstrate the effect of logit tempering. In Figure 12 in Appendix C we report all generalization and fidelity metrics for a range of ensemble sizes. In Appendix A.1 we describe the augmentation procedures in detail.

We first observe that the best augmentation policies for generalization, *MixUp*, and *GAN*[3], are not the best policies for fidelity. Furthermore, although many augmentation strategies enable slightly higher distillation fidelity compared to *Baseline* ($\tau = 1$), even the best augmentation policy, *Mixup* ($\tau = 4$), only achieves a modest 86% test agreement. In fact the *Baseline* ($\tau = 4$) policy is quite competitive, achieving 84.5% test agreement. Many of the augmentation strategies also slightly improve teacher-student KL relative to *Baseline* ($\tau = 4$) (see Figure 12).

Finally, to highlight how poor the student fidelity really is, in Figure 12 (Appendix C) we also report the agreement between the teacher and students distilled from identical, independently trained teacher models, as proposed in Section 3.2. Often these independent students, taught how to mimic a completely different model, have nearly as good test agreement with the teacher as the student explicitly trained to emulate it.

Should data augmentation be close to the data distribution? In theory, *any* data augmentation should help with identifiability: if a student matches a teacher on more data, it is more likely to match the teacher elsewhere. However, the *Noise* and *OOD* augmentation strategies based on noise and out-of-distribution data fail on all metrics, decreasing performance compared to the baseline. In practice, data augmentation has an effect beyond improving identifiability — it has a regularizing effect, making optimization more challenging. We explore this facet of data augmentation in Section 6.

The slight improvements to fidelity with extensive augmentations suggest that increasing the support of the distillation dataset can indeed improve distillation fidelity. However, since the benefit is so small compared to heuristics like logit tempering (which does not modify the

[3]Unlike Figure 1, for Figure 3 we generated new GAN samples every epoch, to mimic data augmentation.

support at all), it is very unlikely that an insufficient quantity of teacher labels is the primary obstacle to high fidelity.

5.2 The data recycling hypothesis

If simply showing the student *more* labels does not always significantly improve fidelity, perhaps we are not showing the student the *right* labels. Additional data augmentation during distillation does give the student more teacher labels to match, but also introduces a distribution shift between the images the teacher was trained on and the images the student is distilling on. Even when the teacher and student have the same augmentation policy, reusing the teacher's training data for distillation violates the assumptions of empirical risk minimization (ERM) because the distillation data is *not* an independent draw from the true joint distribution over images and teacher labels. What if there was no augmentation distribution shift, and the student was distilled on a fresh draw from the joint test distribution over images and teacher labels?

To investigate the effect of recycling teacher data during distillation we randomly split the CIFAR-100 training dataset $\mathcal{D}$ into two equal parts, $\mathcal{D}_0$ and $\mathcal{D}_1$. We train teacher ResNet-56 ensembles on $\mathcal{D}_0$, and then compare s_0, a student distilled on $\mathcal{D}_0$, s_1, a student distilled on $\mathcal{D}_1$ (a dataset unseen by the teacher), and $s_{0\cup1}$, a student distilled on both: $\mathcal{D}_0 \cup \mathcal{D}_1$. Note that the students cannot access the true labels, only those provided by the teacher.

We present the results in Figure 4, varying the ensemble size in the top row and the logit temperature in the bottom row.

Surprisingly, s_0 attains higher test accuracy than s_1, while showing worse ECE and lower fidelity (measured by test teacher-student agreement and test teacher-student KL). Therefore, the hypothesis that s_1 should be a higher fidelity distillation of the teacher than s_0 does hold, but the gain in fidelity *does not* result in s_1 best replicating the teacher's accuracy. The best attributes of s_0 and s_1 are combined by $s_{0\cup1}$, which coincides with how unlabeled data is typically used in practice [1]. The reason for this puzzling observation is simply that for the larger teachers fidelity has not improved *enough* to also improve generalization. In fact, the best teacher-student agreement is only around 85%, no improvement when compared to the results from extensive data augmentation in the last section. We again find that modifying the distillation data can slightly improve fidelity, but the evidence does not support blaming poor distillation fidelity on the wrong choice of distillation data.

6 Optimization: Does the Student Match the Teacher on the Distillation Data?

If poor fidelity is not primarily an identifiability problem from the wrong choice of distillation data, perhaps there is a simpler explanation. Up to this point, we have focused on student fidelity on a held-out test set. Now we turn our attention to student behavior on the distillation data itself. Does the student match the teacher on the data it is trained to match it on?

6.1 More distillation data lowers train agreement

In Figure 1 we presented an experiment distilling ResNet-56 networks on CIFAR-100 augmented with synthetic GAN-generated images. We saw that enlarging the distillation dataset leads to improved teacher-student agreement on test, but the agreement remains relatively low (below 80%) even for the largest distillation dataset that we considered. In Figure 5 (left panel), we report the teacher-student agreement for the same experiment, but now on the distillation dataset. We now observe the opposite trend: as the distillation dataset becomes larger, it becomes more challenging for the student to match the teacher. Even when the student has identical capacity to the teacher, the student only achieves 95% agreement with the teacher when we use $50k$ synthetic images for distillation.

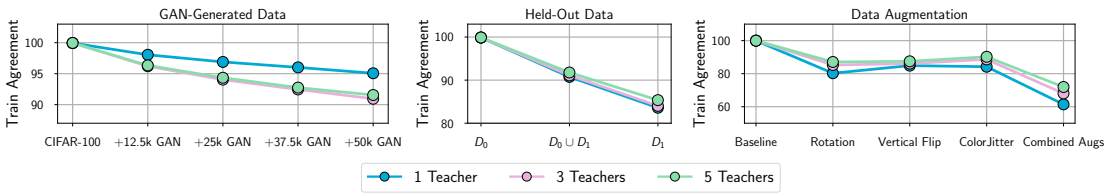


Figure 5: The train agreement for teacher ensembles ($m \in \{1, 3, 5\}$) and student on the distillation data for a ResNet-56 on CIFAR-100 under different augmentation policies. In all panels, increasing the softness of the teacher labels by adding examples not in the teacher train data makes distillation more difficult. **Left**: agreement for the synthetic GAN-augmentation policy from Figure 1. **Middle**: agreement from subsampled CIFAR-100 experiment in Figure 4. **Right**: agreement for some of the augmentation policies in Figure 3. The shaded region is not visible because the variance is very low.

The drop in train agreement is even more pronounced when we use extensive data augmentation. In Figure 5, right panel, we report the teacher-student agreement on the train set with data augmentation for a subset of augmentation strategies presented in Section 5.1. We use the CIFAR-100 dataset and the ResNet-56 model for the teachers and the students (for details, see Section 5.1). In each case, we measure agreement on the augmented training set that was used during distillation. While for the baseline augmentation strategy, we can achieve almost perfect teacher-student agreement, for heavier augmentations the agreement drops dramatically. For the *Rotation*, *Vertical Flip* and *Color Jitter* augmentations, the agreement is between 80% and 90% for all the considered teacher sizes. For *Combined Augs*, the combination of these three augmentation strategies, the agreement drops even further, to just 60% in self-distillation!

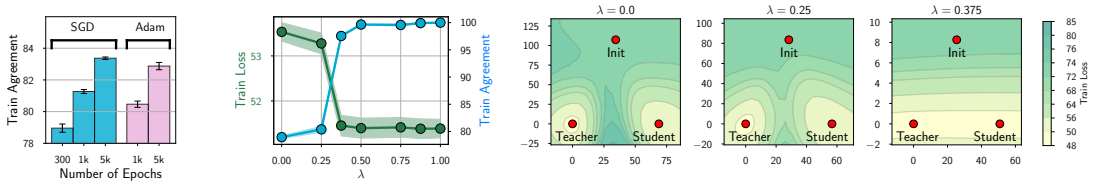
Our intuition about how knowledge distillation should work largely hinges on the assumption that after distillation the student matches the teacher on the distillation set. However, the results presented in this section suggest that in practice the optimization method is unable to achieve high fidelity *even on the distillation dataset* when extensive data augmentation or synthetic data is used. The inability to solve the optimization problem undermines distillation: in order to find a student that would match the teacher on all inputs, we need to at least be able to find a student that would match the teacher on all of the distillation data.

Optimization and the train-test fidelity gap. Notably, despite having the lowest train agreement, the *Combined Augs* policy results in better test agreement than other polices with better train agreement (Figure 3). This result highlights a fundamental trade-off in knowledge distillation: the student needs many teacher labels match the teacher on test, but introducing examples not in the teacher train data makes matching the teacher on the distillation data very difficult.

6.2 Why is train agreement so low?

A simplified distillation experiment. To simplify our exploration, we focus on self-distillation of a ResNet-20 on CIFAR-100. We use the *Baseline* data augmentation strategy, as we found that a ResNet-20 student is unable to match the teacher on train even with basic augmentation. We also replace the BatchNorm layers [22] in ResNet-20 with LayerNorm [2], because we found that with BatchNorm layers even when the teacher and the student have identical weights, they can make different predictions due to differences in the activation statistics accumulated by the BatchNorm layers. Layer normalization does not collect any activation statistics, so the student will match the teacher as long as the weights coincide.

Can we solve the optimization problem better? We verify that the distillation fidelity cannot be significantly improved by training longer or with a different optimizer. By default, in our experiments we use stochastic gradient descent (SGD) with momentum, train the student



(a) Optimizer effect (b) Initialization effect (c) Loss Visualization

Figure 6: **Optimization and distillation**: self-distillation with ResNet-20s with LayerNorm on CIFAR-100. **(a)**: Final train agreement for SGD and Adam optimizers. Training longer improves agreement, but it remains below 85% even after $5k$ epochs. **(b)**: Final train loss and agreement when the initialization is a convex combination of teacher and random weights, $\theta_s = \lambda\theta_t + (1-\lambda)\theta_r$. **(c)**: Projections of the distillation loss surface on the plane intersecting θ_t, the initial student weights, and the final student weights for different λ. When λ is small, the student converges to a suboptimal solution with low agreement. The uncertainty regions correspond to $\mu \pm \sigma$, estimated over 3 trials.

for 300 epochs, and use a weight decay value of 10^{-4}. In Figure 6 we report the results for the SGD and Adam [24] optimizers run for $1k$ and $5k$ epochs without weight decay. Switching from SGD to Adam only reduced fidelity.

For both optimizers, training for more epochs does slightly improve train agreement. In particular, with SGD we achieve 83.3% agreement when training for $5k$ epochs compared to 78.95% when training for 300 epochs. It is possible, though unlikely, that if we train for even more epochs the train agreement could reach 100%. However, training for $5k$ epochs is significantly longer than what is typically done in practice (100 to 500 epochs). Furthermore, the improvement from $1k$ to $5k$ epochs is only about 2%, suggesting that we would need to train for tens of thousands of epochs, even in the optimistic case that agreement improves linearly, in order to get close to 100% train agreement.

The distillation loss surface hypothesis: If we cannot perfectly distill a ResNet-20 on CIFAR-100 with any of the interventions we have discussed so far, we now ask if there is any modification of the problem that *can* produce a high-fidelity student.

In the self-distillation setting, we do know of at least one set of weights that is optimal w.r.t. the distillation loss — the teacher's own weights θ_t. Letting θ_r be a random weight initialization, in Figure 6 (a) we examine the effect of choosing the student initialization to be a convex combination of the teacher and random weights, $\theta_s = \lambda\theta_t + (1-\lambda)\theta_r$. After being initialized in this way, the student was trained as before. Note that $\lambda = 0$ corresponds to a random initialization and $\lambda = 1$ corresponds to initializing the student weights at the teacher weights.

We find that if the student is initialized far from the teacher ($\lambda \leq 0.25$), the optimizer converges to a sub-optimal value of the distillation loss, producing a student that significantly disagrees with the teacher. However at $\lambda = 0.375$ there is a sudden change. The final train loss drops to the optimal value and the agreement drastically increases, and the behavior continues for $\lambda > 0.375$. To further investigate, in Figure 6 (c) we visualize the distillation loss surface for $\lambda \in \{0, 0.25, 0.375\}$ projected on the 2D subspace intersecting θ_t, the initial student weights, and the final student weights. If the student is initialized far from the teacher ($\lambda \in \{0, 0.25\}$), it converges to a distinct, sub-optimal basin of the loss surface. On the other hand, when initialized close to the teacher ($\lambda = 0.375$), the student converges to the same basin as the teacher, achieving nearly 100% agreement.

We have at last identified a root cause of the ineffectiveness of all the previous interventions. Knowledge distillation is unable to converge to optimal student parameters, even when we know a solution and give the initialization a small head start in the direction of an optimum. Indeed, while identifiability can be an issue, in order to match the teacher on all inputs, the student has

to at least match the teacher on the data used for distillation, and achieve a near-optimal value of the distillation loss. In practice, optimization converges to sub-optimal solutions, leading to poor distillation fidelity.

7 Discussion

Our work provides several new key findings about knowledge distillation:

- *Good student accuracy does not imply good distillation fidelity:* even outside of self-distillation, the models with the best generalization do not always achieve the best fidelity.
- *Student fidelity is correlated with calibration when distilling ensembles:* although the highest-fidelity student is not always the most accurate, it is always the best calibrated.
- *Optimization is challenging in knowledge distillation:* even in cases when the student has sufficient capacity to match the teacher on the distillation data, it is unable to do so.
- *There is a trade-off between optimization complexity and distillation data quality:* Enlarging the distillation dataset beyond the teacher training data makes it easier for the student to identify the correct solution, but also makes an already difficult optimization problem harder.

In standard deep learning, we are saved by not needing to solve the optimization problem well: while it true that our training loss is highly multimodal, properties such as the flatness of good solutions, the inductive biases of the network, and the implicit biases of SGD, often enable good generalization in practice. In knowledge distillation, however, good fidelity is directly aligned with solving what turns out to be an exceptionally difficult optimization problem.

Acknowledgements

The authors would like to thank Gregory Benton, Marc Finzi, Sanae Lotfi, Nate Gruver, and Ben Poole for helpful feedback. This research is supported by an Amazon Research Award, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, and NSF 1922658NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science. Samuel Stanton is also supported by a United States Department of Defense NDSEG fellowship.

References

[1] Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? *Advances in neural information processing systems*, 27:2654–2662.

[2] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *Proceedings of the NIPS 2016 Deep Learning Symposium*.

[3] Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., and Kolesnikov, A. (2021). Knowledge distillation: A good teacher is patient and consistent.

[4] Bucilă, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

[5] Che, Z., Purushotham, S., Khemani, R., and Liu, Y. (2015). Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*.

[6] Chebotar, Y. and Waters, A. (2016). Distilling knowledge from ensembles of neural networks for speech recognition. In *Interspeech*, pages 3439–3443.

[7] Chen, D., Mei, J.-P., Wang, C., Feng, Y., and Chen, C. (2020). Online knowledge distillation with diverse peers. In *AAAI*, pages 3430–3437.

[8] Chen, G., Choi, W., Yu, X., Han, T., and Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. In *Advances in neural information processing systems*, pages 742–751.

[9] Cho, J. H. and Hariharan, B. (2019). On the efficacy of knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4794–4802.

[10] Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE.

[11] Fakoor, R., Mueller, J. W., Erickson, N., Chaudhari, P., and Smola, A. J. (2020). Fast, accurate, and simple models for tabular data via augmented distillation. *Advances in Neural Information Processing Systems*, 33.

[12] Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR.

[13] Goldblum, M., Fowl, L., Feizi, S., and Goldstein, T. (2020). Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3996–4003.

[14] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

[15] Gupta, V., Koren, T., and Singer, Y. (2018). Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850.

[16] Han, S., Mao, H., and Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*.

[17] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[18] Heo, B., Lee, M., Yun, S., and Choi, J. Y. (2019). Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787.

[19] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[20] Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. (2016a). Harnessing deep neural networks with logic rules. In *Association for Computational Linguistics*.

[21] Hu, Z., Yang, Z., Salakhutdinov, R., and Xing, E. (2016b). Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679.

[22] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

[23] Kim, Y. and Rush, A. M. (2016). Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

[24] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[25] Krizhevsky, A. et al. (2009). Learning multiple layers of features from tiny images.

[26] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989a). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

[27] LeCun, Y., Denker, J. S., Solla, S. A., Howard, R. E., and Jackel, L. D. (1989b). Optimal brain damage. In *NIPs*, volume 2, pages 598–605. Citeseer.

[28] Liu, X., Wang, X., and Matwin, S. (2018). Improving the interpretability of deep neural networks with knowledge distillation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 905–912. IEEE.

[29] Malinin, A., Mlodozeniec, B., and Gales, M. (2019). Ensemble distribution distillation. In *International Conference on Learning Representations*.

[30] Marcel, S. and Rodriguez, Y. (2010). Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488.

[31] Mariet, Z., Jenatton, R., Wenzel, F., and Tran, D. (2020). Distilling ensembles improves uncertainty estimates. *Proceedings of the 3rd Symposium on Advances in Approximate Bayesian Inference*, pages 1–10.

[32] Meng, Z., Li, J., Gong, Y., and Juang, B.-H. (2018). Adversarial teacher-student learning for unsupervised domain adaptation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5949–5953. IEEE.

[33] Mishra, A. and Marr, D. (2018). Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *International Conference on Learning Representations*.

[34] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

[35] Mobahi, H., Farajtabar, M., and Bartlett, P. L. (2020). Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*.

[36] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

[37] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE.

[38] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[39] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2015). Fitnets: Hints for thin deep nets. *International Conference on Learning Representations*.

[40] Shen, Z., He, Z., and Xue, X. (2019). Meal: Multi-model ensemble via adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4886–4893.

[41] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*.

[42] Tan, S., Caruana, R., Hooker, G., and Lou, Y. (2018). Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310.

[43] Tokozume, Y., Ushiku, Y., and Harada, T. (2018). Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494.

[44] Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., and Richardson, M. (2017). Do deep convolutional nets really need to be deep and convolutional? *International Conference on Learning Representations*.

[45] Yim, J., Joo, D., Bae, J., and Kim, J. (2017). A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141.

[46] You, S., Xu, C., Xu, C., and Tao, D. (2017). Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294.

[47] Zagoruyko, S. and Komodakis, N. (2017). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *International Conference on Learning Representations*.

[48] Zeng, X. and Martinez, T. R. (2000). Using a neural network to approximate an ensemble of classifiers. *Neural Processing Letters*, 12(3):225–237.

[49] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*.

Appendix

Outline:

A. Implementation details for all experiments.

B. Further experiments for VGG distillation and ensemble distillation.

C. Experiments addressing spurious explanations for poor student fidelity.

A Implementation Details

Here we briefly describe key implementation details to reproduce our experiments. Data augmentation details are given in A.1, followed by architecture details in A.2, and finally training details are provided in A.3. The reader is encouraged to consult the included code for closer inspection.

A.1 Data Augmentation Procedures

Some of the data augmentation procedures we consider attempt to generate data that is close to the train data distribution (standard augmentations, GAN, mixup). Others (random noise, out-of-domain data) produce data for distillation that the teacher would never encounter during training. In particular, we compare the following augmentation procedures:

Baseline Augmentations As a baseline, we use the same data augmentation strategy that was used to train the teachers during distillation: we apply random horizontal flips ($p = 0.5$) and random shifts via pad/random-crop with a 4 pixel pad width. In all of the configurations we consider in this section we use this set of augmentations along with other strategies, unless stated otherwise.

Conventional Image Transformations Standard data augmentations used in computer vision [30]: random rotations by up to 20 degrees, random vertical flips, color jitter and all possible combinations. For more detail on the parameters of the augmentations, see Appendix A.1.

Mixup Mixup is an effective regularization technique originally proposed to increase generalization and robustness of deep networks [49, 43]. Instead of training on original dataset, the network is trained on convex combination of images with targets mixed in the same way. We adapt mixup to knowledge distillation as follows: on each iteration we construct random pairs of objects x, x' from the training set and mix them as $\lambda \cdot x + (1 - \lambda) \cdot x'$, where the coefficient λ is sampled uniformly on $[0, 1]$[4].

Synthetic GAN-generated images We use a Spectral Normalization GAN (SN-GAN) trained on CIFAR-100 [34] to generate synthetic data for distillation. We used the same pretrained SN-GAN (FID = 74.2617, IS = 6.6023) for all experiments. Our training procedure and sample images are provided in Appendix A.1. Our synthetic augmentation procedure was the following: for each minibatch of real training data, we concatenated synthetic images sampled from a pretrained SN-GAN at a ratio of 1 synthetic image to 4 real images.

Random noise To observe the effect of unnatural images in the distillation dataset we augment with images sampled pixel-wise from uniform $[0, 1]^d$. During distillation each image in a minibatch is randomly resampled with probability $p = 0.2$.

Out-of-domain data Finally, we consider using images from the SVHN dataset [36] which is semantically unrelated to the target CIFAR-100 dataset.

[4]Note that unlike in the original mixup procedure we are only mixing the inputs and we use the predictions of the teacher on the mixed inputs as the target for the student.

We use the `torchvision.transoforms` package [38] to implement the augmentations from the Baseline Augmentations and Conventional Image Transformations categories:

- Horizontal flips: `torchvision.transforms.RandomHorizontalFlip()`
- Random shifts: `torchvision.transforms.RandomCrop(size=<input_size>, padding=4)`
- Vertical flips: `torchvision.transforms.RandomVerticalFlip()`
- Color jitter: `torchvision.transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.2)`
- Random rotations: `torchvision.transforms.RandomRotation(degrees=20)`

A.2 Network Architectures

Image Classifiers For experiments on CIFAR-100 we used preactivation ResNets with batchnorm, skip connections [17], and the standard three-stage macro-structure, varying the number of layers in each stage (i.e. the depth of the network). For all choices of depth we used the same number of filters in each stage (16, 32, and 64, respectively). In Section B.1 we use a VGG-16 network without batch-normalization, with implementation directly adapted from `https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py`. For experiments on MNIST/EMNIST we used a 5-layer LeNet [26].

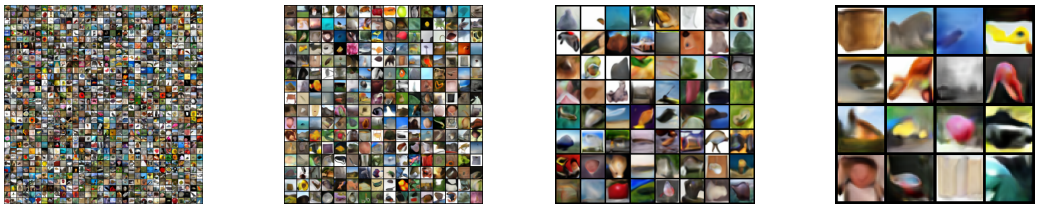


Figure 7: Sample images from our SN-GAN (FID = 74.2617, IS = 6.6023) trained on CIFAR-100.

Image Generators We used the standard three-stage ResNet architectures for the SN-GAN generator and discriminator from Miyato et al. [34]. The generator latent dimension was 128, and each generator stage had 256 filters. The discriminator had 128 filters in each stage. Sample images are shown in Figure 7.

A.3 Training Procedures

In our experiments, we consider pre-activation ResNet networks with depths 20, 56 and 110 [17]. We evaluate on MNIST/EMNIST [10] and CIFAR-100 [25], focusing primarily on the latter. We chose CIFAR-100 rather than CIFAR-10 because increasing the problem difficulty increases the gap in performance between a single model and an ensemble, making significant trends more apparent. We independently train each network in the teacher-ensemble to minimize $\mathcal{L}_{\text{NLL}}$ for 200 epochs, and we distill each student by training it to minimize $\mathcal{L}_{\text{KD}}$ for 300 epochs (i.e. we take α in $\mathcal{L}_s$ to be 0). Note that in the literature one typically sees $\alpha > 0$. We have chosen $\alpha = 0$ so that our objective reflects our aim of producing the highest fidelity student possible. We use an SGD optimizer with initial learning rate 5×10^{-2} and cosine annealing learning rate schedule (for further details see Appendix A). We produce augmented datasets for distillation by sampling images from a set of specified sources, including rotations or color jitter applied to ground truth images, uniform 'white noise' images, or synthetic GAN-generated images. Unless specified otherwise, the only augmentations applied when training the teacher were the standard

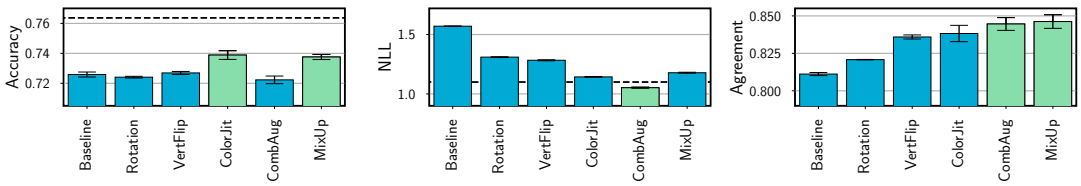


Figure 8: Test accuracy, negative log-likelihood and teacher-student agreement when distilling a 5-component VGG-16 teacher ensemble into a VGG-16 student on CIFAR-100 with varying augmentation policies. The best performing policy is shown in green, results averaged over 3 runs and error bars indicate $\pm\sigma$. The results are generally analogous to the results for the ResNet-56 architecture reported in Section 5.1. *MixUp* and *ColorJit* provide the best student accuracy, while *CombAug* provides the best NLL. *CombAug* and *MixUp* provide the best teacher-student agreement.

random horizontal flip ($p = 0.5$) and padded random crop (4 pixel pad width), regardless of the choice of distillation dataset.

Teacher Image Classifiers: The teacher models were trained through the standard empirical cross-entropy loss for 200 epochs with a batch size of 256 using SGD with momentum (0.9 momentum weight) and weight decay of 1.0×10^{-4}. We used a cosine annealing learning rate schedule with $\eta_{\max} = 0.1$, $\eta_{\min} = 0$. For data augmentation we used random horizontal flips ($p = 0.5$) and random crops (padding width 4).

Student Image Classifiers: Our student models were distilled through the temperature-scaled teacher-student cross-entropy with varying temperatures τ for 300 epochs, with a batch size of 128 using SGD with Nesterov momentum (0.9 momentum weight) and weight decay of 1e-4. We used a cosine annealing learning rate schedule with $\eta_{\max} = 5.0 \times 10^{-2}$, $\eta_{\min} = 1.0 \times 10^{-4}$. For details on the data augmentation procudures we considered, the reader is directed to Appendix A.1.

Image Generators: For synthetic image generation we trained SN-GAN models with the hinge discriminator loss from Miyato et al. [34]. We trained the generator for 100K gradient steps with a batch size of 128. For each generator step, we took 5 discriminator steps. We used the hinge discriminator loss. We used Adam ($\beta_1 = 0, \beta_2 = 0.9$) and a linearly decayed learning rate $\eta_{\max} = 2.0 \times 10^{-4}$, $\eta_{\min} = 1.0 \times 10^{-6}$. We used random horizontal flips ($p = 0.5$) as data augmentation for the discriminator. To evaluate FID and IS scores, we used 5K samples from the generator and the pretrained PyTorch Inception-v3 networks[5]. For the discriminator and generator architectures the reader is referred to Appendix A.2.

B Additional Experimental Results

In section B.1 we present the results of distilling VGG networks to verify that poor distillation fidelity is not restricted to ResNets. In section B.2 we explore the qualitative effect of teacher ensemble size and the component depth on the predictive distributions on train and test, to get a better understanding of what the students are being asked to emulate.

B.1 Distilling VGG Networks

As we noted in Section 4.4 in the main text, there was the possibility that our initial findings on the difficulty of achieving high-fidelity students with ResNets on CIFAR-100 could be attributed

[5] https://pytorch.org/docs/stable/torchvision/models.html?highlight=inception#torchvision.models.inception_v3

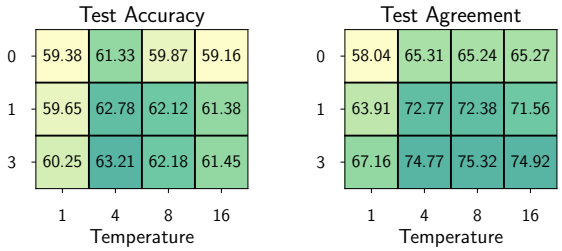


Figure 9: Subsampled CIFAR-100 experiment performed with ResNet20 networks. ResNet20 networks are much less confident on train than ResNet56 networks. As a result increasing the ensemble size will improve the student even with a small temperature setting $\tau = 1$.

to the choice of architecture. ResNet-style backbones are ubiquitous across most computer vision tasks, so even were the issue restricted to ResNets it would merit close investigation. Nevertheless, in the interest of empirical rigor we repeat the augmentation ablation in Section 5.1 with VGG networks and a subset of the augmentation policies.

In Figure 8, the teacher is a 5-component ensemble of VGG-16 networks trained with the *Baseline* augmentation policy (horizontal flips and random crops). We report the student accuracy, negative log-likelihood and teacher-student agreement for a VGG-16 student trained with different data augmentation policies.

The results are generally analogous to the ones for ResNet-56 presented in Section 5.1. The *CombAug* augmentation strategy underperforms all other strategies, including *Baseline*, on student accuracy, but provides the best results on NLL and only slightly loses to *MixUp* on teacher-student agreement. This result again highlights that the best augmentation policies for generalization do not necessarily provide the best distillation fidelity. Finally, regardless of the augmentation strategy, the agreement on test does not exceed 85%.

B.2 Understanding Ensemble Distillation

Since deep ensemble components are typically large networks that achieve almost 100% accuracy on train with very high confidence, it is tempting to assume that each ensemble component conveys effectively the same information when used for distillation. One consequence of that assumption would be that adding ensemble components would produce little or no improvement in the student if the distillation was performed on train.

In fact, we find that although the component networks are indeed very confident on the train, there is sufficient variation in their predictive distributions for the student to benefit significantly (see Appendix B.2 for more discussion). As discussed in Section 3.1 In order to see this benefit, one must choose τ large enough that the student directs some capacity towards mimicking the smaller teacher logits (Figure 4, bottom). If τ is chosen too small (e.g. $\tau = 1$), then the student distilled from a 3-component ensemble is no better than a student distilled from a single network. The improvements in student performance and fidelity taper off fairly quickly as ensemble components are added.

The correct choice of τ depends on the level of confidence the teacher has on train. ResNet56 networks achieve nearly 100% accuracy on train with high confidence, so a temperature like $\tau = 4$ works well. When ResNet20 networks are used (networks which are not capable of perfectly fitting CIFAR-100), we see that lower temperatures can be used, although $\tau = 4$ still outperforms other choices (Figure 9).

In Figure 10 in blue we report the performance of a ResNet-56 student distilling an ensemble of k networks of the same architecture as we vary k. In the main text we only considered

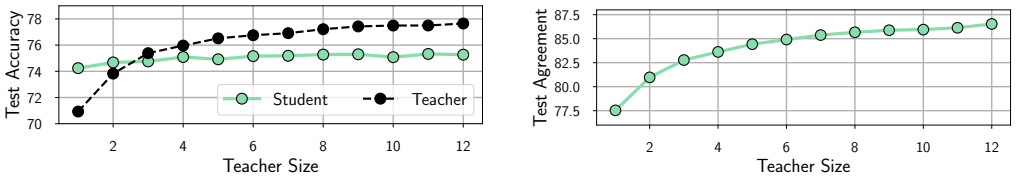
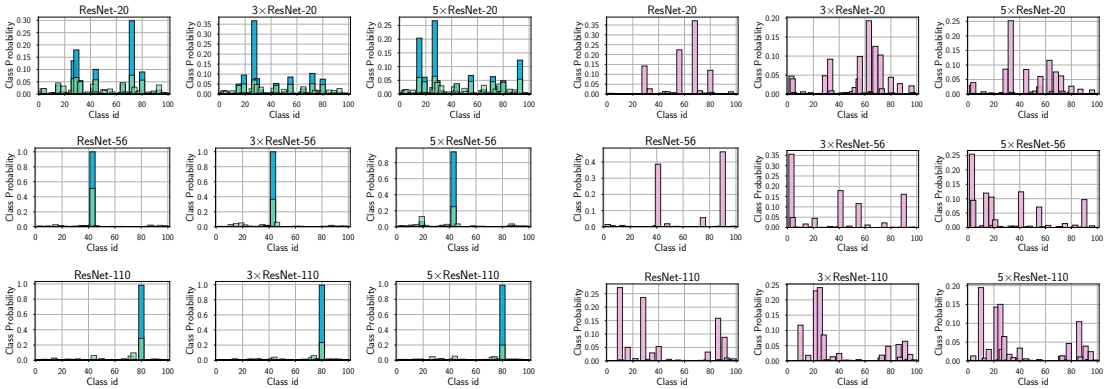


Figure 10: The effect on accuracy **(left)** and agreement **(right)** of the number of models (k) in the teacher ensemble ($\alpha = 0,\ \tau = 1$). Student accuracy quickly saturates as k increases, despite continuing improvements in teacher accuracy. The teacher-student agreement continues to improve after the accuracy has saturated.



(a) Train Data (b) Test Data

Figure 11: Teacher predictive distributions for example images from CIFAR-100 train **(left)** and test **(right)**. For train examples we show the distribution when $\tau = 1$ in blue and the tempered distribution when $\tau = 4$ in green. For test examples we only show $\tau = 1$. Each row corresponds to a different teacher depth, and the column corresponds to the number of ensemble components.

teacher-ensembles with up to 5 components – here we provide results for up to 12 components. In agreement with the results on self-distillation [12, 35], we see a significant improvement over the teacher at $k = 1$ and $k = 2$. However, surprisingly, the performance of the student does not substantially improve as we increase the number k of models in the teacher ensemble past 2, even though the performance of the teacher improves monotonically with k. Interestingly, the matching between the teacher and the student also increases monotonically with k.

In Figure 12 we report more detailed results for the experiment in Figure 3 (in the main text). In particular, for the sake of simplicity we only reported results for $k = 5$ in the main text. Here we report results for $k = 1$ and $k = 3$ as well for comparison.

Figure 11 provides sample teacher-ensemble predictive distributions for images drawn from CIFAR-100 train and test, varying the number of teacher components, the teacher component depth, and the logit temperature.

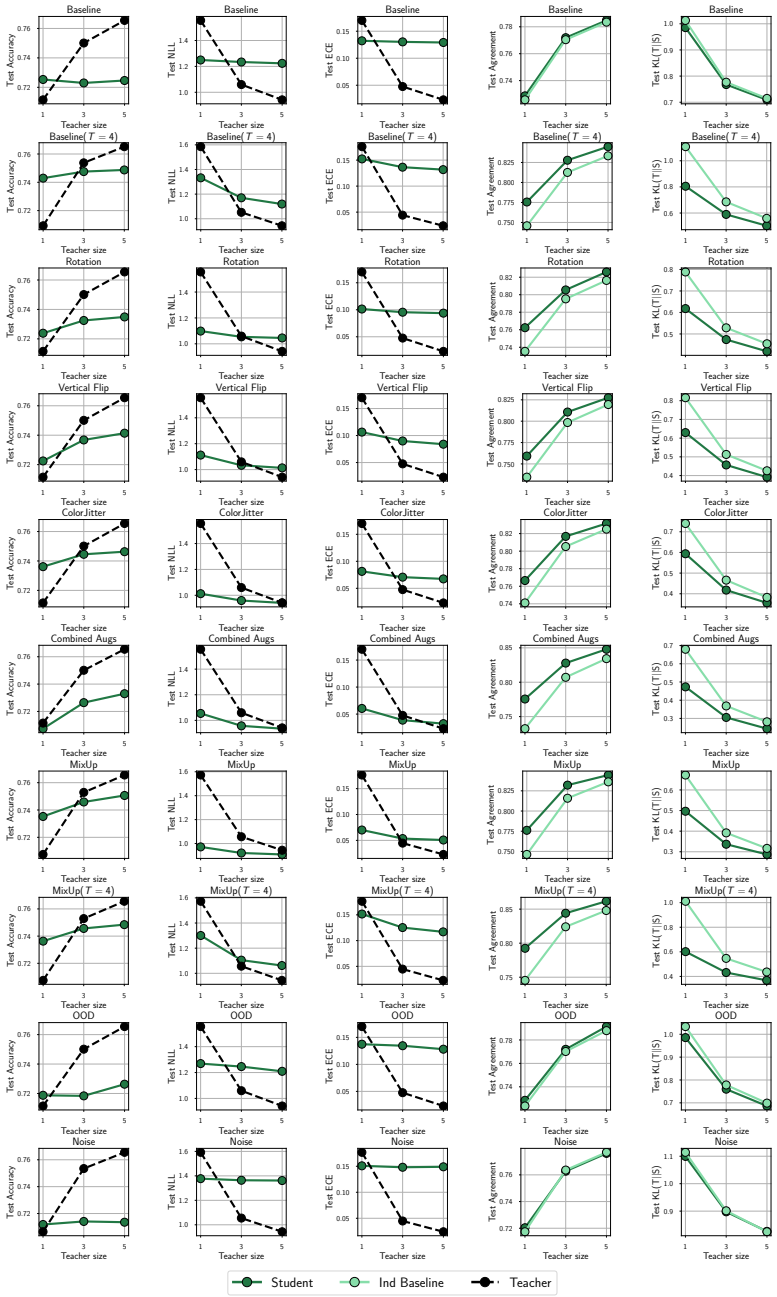


Figure 12: Detailed results for the experiment in Section 5.1. Each row corresponds to a different augmentation procedure, and each column is a different evaluation metric. Notably, we see that the student distilled with mixup and $\tau = 1$ is the best overall in terms of NLL (though not test accuracy) beating even the teacher-ensemble for all values of k. The independent baseline serves as a reference to aid in the interpretation of fidelity metrics.

C Addressing Other Possible Causes of Poor Fidelity

In section C.1 we demonstrate that increasing student capacity does not substantially improve fidelity. In section C.2 we demonstrate that the common practice of showing the student both the real labels (when available) and the teacher labels tends to decrease fidelity. In section C.3 we demonstrate that although ensembles with more components are larger models, they are easier to emulate.

C.1 Ablating Student Capacity

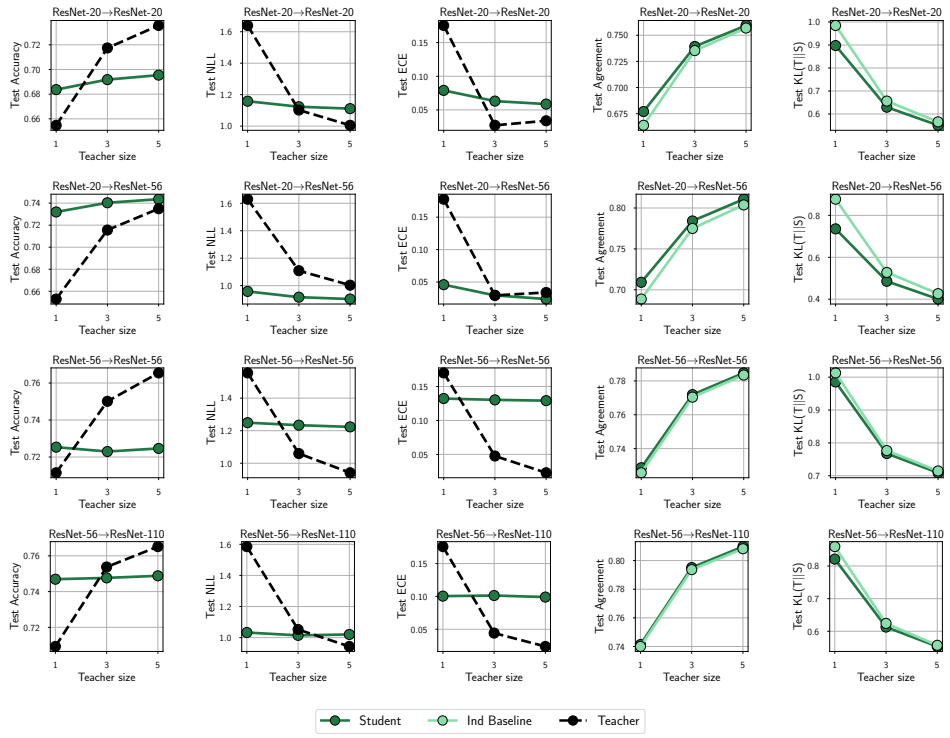
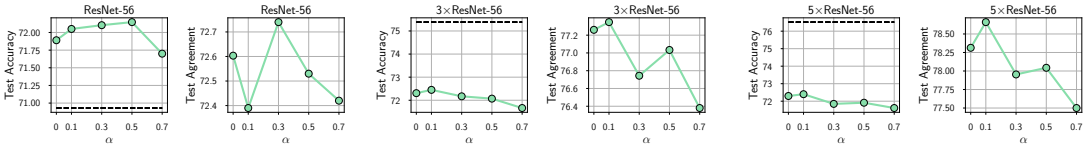


Figure 13: Here we show the effect of increasing the student capacity, holding the teacher capacity fixed. The top two rows correspond to ResNet20 teacher-ensemble components with ResNet20 and ResNet56 students, respectively. The bottom two rows are similarly ResNet56 teacher components with ResNet56 and ResNet110 students. The column corresponds to the evaluation metric. Increasing student capacity from 20 to 56 provides some benefit to both accuracy and fidelity, but increasing student capacity from 56 to 110 improves only accuracy.

As we noted in Section 4.4 in the main text, another possible cause of our initial result observing the difficulty of high fidelity distillation of ensembles of ResNets on CIFAR-100 is that a single student network does not have *capacity* to perfectly emulate an ensemble of multiple networks. This explanation is already rendered unlikely by our similar observations in the context of self-distillation. Nevertheless for completeness in Figure 13 we demonstrate the effect of increasing student capacity beyond that of the individual teacher components. Increasing the student capacity does slightly improve fidelity – doubling the student network depth results in a 2% to 3% improvement in test agreement.



(a) 1 Teacher (b) 3 Teachers (c) 5 Teachers

Figure 14: Results ablating α ($\tau = 1$). Taking $\alpha > 0$ can improve student accuracy in the self-distillation regime, but does not consistently improve teacher-student agreement. When $k > 0$ there is a slight benefit at $\alpha = 0.1$, after which the effect is negative for both accuracy and agreement.

C.2 Does Showing the Student Ground Truth Labels Make a Difference?

In Figure 14 we investigate the effect of the relative weight of the distillation loss terms $\mathcal{L}_{\text{NLL}}$ and $\mathcal{L}_{\text{KD}}$ when distilling teacher-ensembles with ResNet56 components into a ResNet56 student on CIFAR-100 with $\tau = 1$. We observe that in the self-distillation regime taking $\alpha > 0$ improves test accuracy, but not test agreement. When $k > 0$, there is a slight benefit when $\alpha = 0.1$, but for most values tried the effect was deleterious to both accuracy and fidelity.

C.3 Ablating the Size of the Teacher Ensemble

In Figure 10 we demonstrate the effect of increasing the number of teacher ensemble components on test accuracy and agreement. Although it is plausible that ensembles with more components would have more complex predictive distributions that would be difficult for a single student to match, in reality we see the exact opposite. Deep ensembles with *more* components are easier to emulate. One possible explanation is that adding more ensemble components smooths the logits of unlikely classes, making the distribution easier to match. Closer investigation into this phenomenon could potentially yield insights into how to improve distillation fidelity in general. Unfortunately the increase in teacher-student agreement does not seem to translate into better student accuracy, since we observe the student accuracy plateauing around 3 or 4 ensemble components.