Does Knowledge Distillation Really Work?

Anonymous Author(s)
Affiliation
Address
email

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

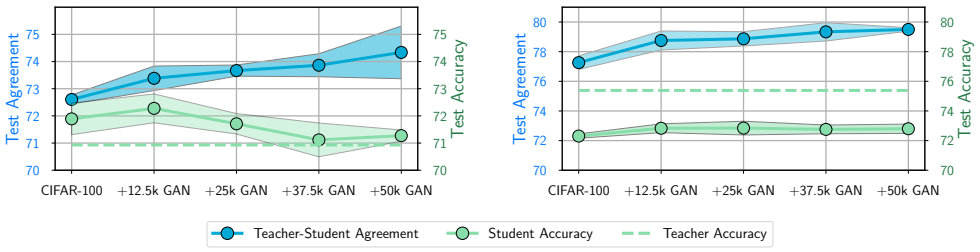
Large, deep networks can learn representations that generalize well. While smaller, more efficient networks lack the *inductive biases* to find these representations from training data alone, they may have the *capacity* to represent these solutions [e.g., 1, 14, 22, 30]. Influential work on *knowledge distillation* [16] argues that Bucilă et al. [3] "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 [11, 28]. 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.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(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 [26, 30], unsupervised domain adaptation [25], improved object detection [7], model transparency [32], and adversarial robustness [12, 29]. Seminal work by Bucilă et al. [3] 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 [37], an idea which resonates with more recent work on the distillation of deep ensembles [1, 5, 33, 35].

Recently Fakoor et al. [10] 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. [24] 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, de-coupling 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.

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. [16] 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,

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

$\mathcal{L}_{\text{NLL}}$ 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 ($\hat{p}_s$). $\mathcal{L}_{\mathrm{KD}}$ 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 $\mathcal{L}_{\mathrm{KD}}$ is similarly equivalent to the KL from the teacher to the student, $\mathrm{KL}(\hat{p}_t || \hat{p}_s)$. 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 $\tau \to +\infty$, $\nabla_{\mathbf{z}_s} \mathcal{L}_{\mathrm{KD}}(\mathbf{z}_s, \mathbf{z}_t) \approx \mathbf{z}_t - \mathbf{z}_s$, and thus in the limit $\nabla_{\mathbf{z}_s} \mathcal{L}_{\mathrm{KD}}$ 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) [13]. To measure fidelity, we report the following:

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

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

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 [9]. The EMNIST train set contains 697,932 images.

[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.

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 [27] 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.

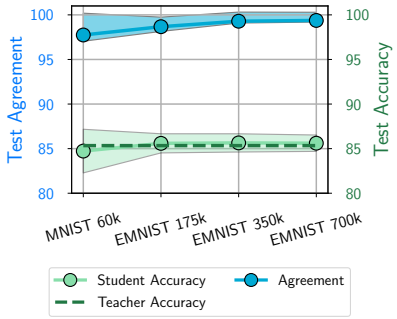


Figure 2: LeNet-5 self-distillation on MNIST with additional distillation data. The shaded region corresponds to $\mu \pm \sigma$, estimated over 3 trials.

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.

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. [11]. 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 already 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., 8, 15, 20, 34, 36]. 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 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.

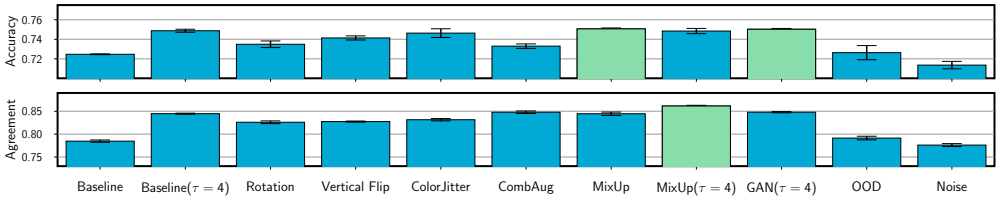


Figure 3: **Data augmentation and distillation**: Test accuracy and teacher-student agreement when distilling a 5-component ResNet-56 teacher ensemble into a ResNet-56 student on CIFAR-100 with varying augmentation policies. The best performing policy is shown in green, results averaged over 3 runs. Additional metrics are reported in Figure 11 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 $\pm\sigma$.

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., 17, 18, 4, 23, 6]. 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. 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.

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 [31] 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?

[2]See, for example, Lemma 1 in Fakoor et al. [10].

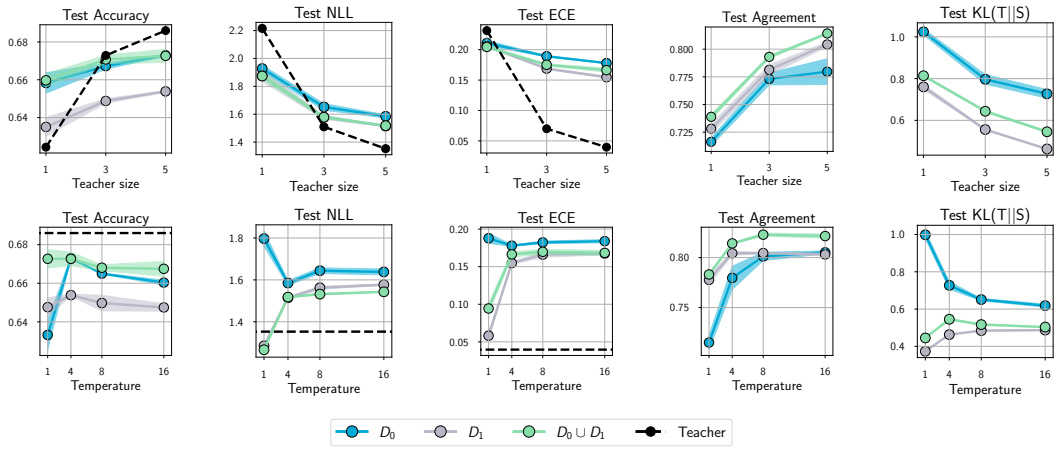


Figure 4: **Data recycling and distillation**: results on subsampled CIFAR-100. **Top:** We fix the temperature ($\tau = 4$) and vary the number of ensemble components (m), comparing students distilled on the same dataset as the teacher ($\mathcal{D}_0/\mathcal{D}_0$), a reserved dataset ($\mathcal{D}_0/\mathcal{D}_1$), 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.

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 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 11 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 11).

Finally, to highlight how poor the student fidelity really is, in Figure 11 (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

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

small compared to heuristics like logit tempering (which does not modify the 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 the original $\mathcal{D}_0$, s_1, a student distilled on the unseen $\mathcal{D}_1$, and $s_{0\cup 1}$, 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 $s_{0\cup 1}$, 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 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.

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.

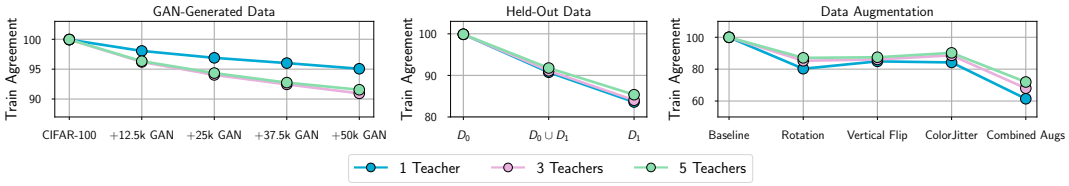


Figure 5: The train agreement for teacher ensembles ($m \in \{1, 3, 5\}$) and student on the distillation data for a ResNet-56 on CIFAR-100 under different augmentation policies. In all panels, increasing the softness of the teacher labels by adding examples not in the teacher train data makes distillation 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.

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.

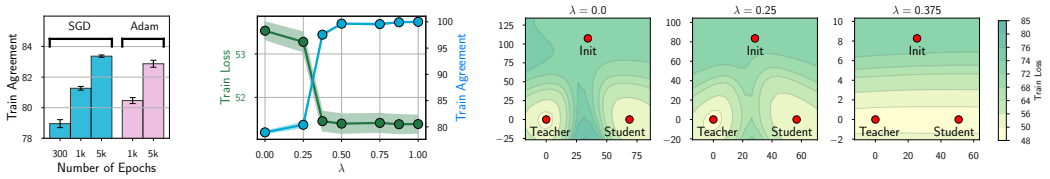
6.2 Why is train agreement so low?

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 [19] 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 for 300 epochs, and use a weight decay value of 10^{-4}. In Figure 6 we report the results for the SGD and Adam [21] 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.



(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, θ_r, and θ_s for different λ. When λ is small, the student converges to a suboptimal solution with low train agreement. The shaded region corresponds to $\mu \pm \sigma$, estimated over 3 trials.

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 provide visualizations of the distillation loss surface for $\lambda \in \{0, 0.25, 0.375\}$ projected on the 2D subspace intersecting θ_t, θ_r, and θ_s. 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.

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. *arXiv preprint arXiv:1607.06450*.

[3] 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.

[4] 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*.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] Fakoor, R., Mueller, J., Erickson, N., Chaudhari, P., and Smola, A. J. (2020). Fast, accurate, and simple models for tabular data via augmented distillation. *arXiv preprint arXiv:2006.14284*.

[11] Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born again neural networks. *arXiv preprint arXiv:1805.04770*.

[12] 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.

[13] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*.

[14] Han, S., Mao, H., and Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

[15] 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.

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

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

[18] 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.

[19] 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.

[20] 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.

[21] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

[23] 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.

[24] Malinin, A., Mlodozeniec, B., and Gales, M. (2019). Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*.

[25] 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.

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

[27] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

[28] Mobahi, H., Farajtabar, M., and Bartlett, P. L. (2020). Self-distillation amplifies regularization in hilbert space. *arXiv preprint arXiv:2002.05715*.

[29] 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.

[30] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

[31] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[32] 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.

[33] Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., and Richardson, M. (2016). Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*.

[34] 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.

[35] 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.

[36] Zagoruyko, S. and Komodakis, N. (2016). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] (see Appendix D.1)
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] (see Appendix D.2)
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] (see Appendix D.3
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] (see Appendix D.4)
   (b) Did you mention the license of the assets? [Yes] (see Appendix D.4)
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]