# Testing knowledge distillation theories with dataset size

**Giulia Lanzillotta**[1,2], **Felix Sarnthein**[3], **Gil Kur**[2], **Thomas Hofmann**[2], **and Bobby He**[2]

[1]ETH AI Center, Switzerland
[2]Department of Computer Science, ETH Zurich, Switzerland
[3]ELLIS Institute Tübingen, Germany

## Abstract

The concept of knowledge distillation (KD) describes the training of a student model with a teacher model and is a widespread technique in deep learning. However, it is still not clear how and why distillation works. Previous studies focus on two central aspects of distillation: *model size*, and *generalisation*. In this work we study distillation in a third dimension: dataset size. We present a suite of experiments across a wide range of datasets, tasks and neural architectures, and consistently observe that the gap in test error between distillation and the standard label training is increased as the dataset size is reduced. We call this newly discovered property the *data efficiency* of distillation. Equipped with this new perspective, we test the predictive power of existing theories of KD as we vary the dataset size. Our results disprove the hypothesis that distillation can be understood as label smoothing, and provide further evidence in support of the dark knowledge hypothesis. Ultimately, this work reveals that the dataset size may be a fundamental but overlooked variable in the mechanisms underpinning distillation.

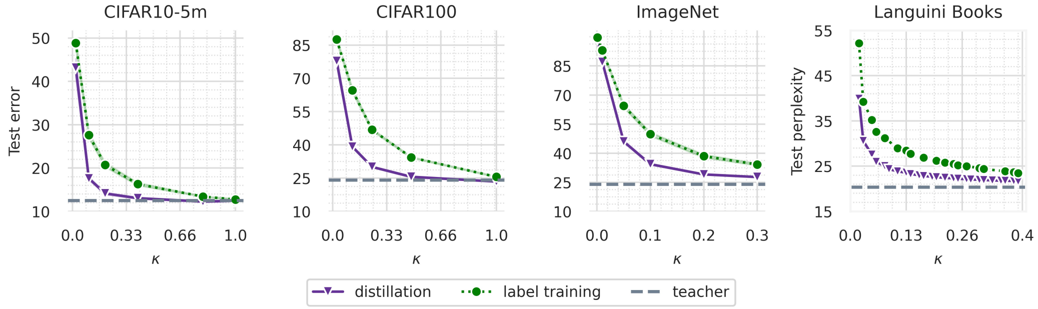

Figure 1: **Distillation is data efficient.** Test error (for image classification) and perplexity (for autoregressive language modelling) as a function of the relative training dataset size $\kappa$, averaged over $5$ seeds. We compare models of the same architecture trained with either label training or knowledge distillation. We observe that distillation dominates label training in the low data regimes.

## 1 Introduction

Knowledge distillation (KD) was introduced by Buciluǎ et al. (2006); Hinton et al. (2015) as a way to *transfer knowledge* between two models with potentially different parameterisations. In its simplest form, it consists of replacing the targets in the loss function with the outputs of a second model, known as the *teacher model*. Several variants of the original formulation have been proposed, and

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

KD is now a widespread technique in deep learning (Zagoruyko & Komodakis, 2016; He & Ozay, 2021; Touvron et al., 2021a; Caron et al., 2021; Beyer et al., 2022).

The study of distillation has so far focused on two central aspects thereof: model size and generalisation performance. Generally, it has been observed that through distillation one can reduce the model size by a considerable factor without harming performance. And, more remarkably, the student can generalise better than a teacher of the same size (a setting known as *self-distillation*), even without any additional labels (Furlanello et al., 2018). All these observations were collected by training the student and the teacher on the same amount of data.

In this work we look at distillation in relation to the dataset size. In doing so we discover an *inherent and unique characteristic of distillation*, a phenomenon which we call the "*data efficiency of distillation*". In a nutshell, we observe that the performance gains which have been historically linked to self-distillation are more pronounced as the dataset size is reduced and they extend beyond the self-distillation setting. This effect is consistent over a wide variety of experiments including convolutional networks & transformers trained on vision & language data (see Figure 1). Existing theoretical work has observed that distillation offers statistical efficiency in fixed features settings (i.e. either linear models or neural networks in the NTK regime) (Phuong & Lampert, 2019; Ji & Zhu, 2020; Panahi et al., 2022; Zhao & Zhu, 2023; Menon et al., 2021). To the best of our knowledge, we are the first to empirically corroborate these observations on popular neural networks and benchmarks.

As of today, there is no comprehensive theoretical account of distillation and the research community is divided between several existing narratives. We revisit hypotheses on label smoothing (Yuan et al., 2020; Zhou et al., 2021), distillation fidelity (Stanton et al., 2021), and feature learning (Allen-Zhu & Li, 2020; He & Ozay, 2021) in light of this new data-centric perspective. Our work contributes to the *understanding* of distillation by presenting new evidence and, more generally, a new perspective which can expose the empirical bias in the current theories, helping to falsify or confirm them.

## 2 The data efficiency of distillation.

**Experimental Setup.** We adopt a simple experimental setup to compare distillation and label training. We start with a trained teacher and train two equivalent models using either the teacher logits (distillation, KD) or labels (label training, LT) as targets, obtaining a *student pair* $(p_{KD}, p_{LT})$. For a teacher trained on $N$ samples, and a training dataset of $M$ samples, we denote by $\kappa := M/N$ the *fraction of training data* relative to the teacher, and we repeat the experiments for different $\kappa$ in the range $(0, 1)$. Additionally, we include a temperature hyperparameter in the distillation loss as Hinton et al. (2015), which we finetune on the dataset. We present experiments on both image classification, as is more common in the distillation literature, and also autoregressive language modelling tasks. A complete description of the datasets and training procedure can be found in App. D. As metrics, we use *test error* or *test accuracy* $Acc(\cdot)$ for vision, and *test perplexity* $PPL(\cdot)$ for text data.

**Results.** In Figure 1, we compare distillation to label training as a function of the data fraction $\kappa$ across datasets, tasks and architectures. Surprisingly, we observe in all settings that distillation outperforms label training when $\kappa < 1$. Moreover, the gap in *test* performance -hereafter called the *performance increment* (PI)- peaks at low values of $\kappa$ (i.e. $0.05$ to $0.3$ for the datasets tested).

To better appreciate this finding it is worth to place it in the context of the existing literature on distillation. It is widely reported that *self-distillation* leads to increased generalisation (Furlanello et al., 2018; Allen-Zhu & Li, 2020; Stanton et al., 2021). Since typically student and teacher are trained with the same amount of data, this observation corresponds to the $\kappa = 1$ slice in Figure 1. Now compare the $0.20\%$ and $1.3\%$ PI reported by Furlanello et al. (2018) respectively on CIFAR10 and CIFAR100 with the $10\%$ and $25\%$ PI which we observe (annotated on Figure 1). Additionally, on ImageNet and on Languini Books we observe PIs around $15\%$ and $10\%$, respectively. In short, we discover that the generalisation boost of distillation is *not only preserved but amplified* as the fraction of training data $\kappa$ is reduced, an effect which we call *data efficiency of distillation*.

## 3 Existing theories of distillation.

As mentioned above, multiple hypotheses have been produced to explain how distillation works, each presenting empirical evidence in its own support. In this section, we reproduce the respective experiments in our setup in order to test which of these intuitions also apply to the $\kappa \neq 1$ case, and which are an artifact of the so far limited perspective.

### 3.1 Label smoothing

|  | CIFAR100 |  |  | CIFAR10 |  |  |
|---|---|---|---|---|---|---|
| $\kappa$ | LT | LS | KD | LT | LS | KD |
| 0.02 | 12.44±0.81 | +0.48 ± 0.49 | +9.77 ± 1.10 | 56.92 ± 0.46 | −0.66 ± 0.53 | +4.74 ± 0.86 |
| 0.1 | 35.36 ± 0.84 | +0.38 ± 0.68 | **+25.46** ± 0.76 | 74.20 ± 0.28 | −0.84 ± 0.31 | **+6.01** ± 0.33 |
| 0.2 | 53.21 ± 0.44 | +0.48 ± 0.68 | +16.72 ± 0.53 | 78.82 ± 0.47 | −0.22 ± 0.44 | +4.80 ± 0.65 |
| 0.4 | 65.66 ± 0.24 | **+0.60** ± 0.51 | +8.75 ± 0.54 | 82.35 ± 0.28 | −0.16 ± 0.38 | +3.26 ± 0.30 |
| 1.0 | 74.42 ± 0.22 | +0.47 ± 0.41 | +2.12 ± 0.24 | 85.43 ± 0.15 | **+0.28** ± 0.23 | +1.53 ± 0.24 |

Table 1: **Distillation is data efficient, label smoothing is not.** Classification accuracy of label training (LT), and PI of label smoothing (LS) and knowledge distillation (KD) on CIFAR10 and CIFAR100.

Yuan et al. (2020); Zhou et al. (2021) propose that the better generalisation of distillation is a result of the regularization effect of smooth labels. In other words, the class relationship structure implicit in the teacher network's output is not the cause of distillation's performance but rather what matters is non-zero probability mass on non-target classes.

We are interested in asking whether the label smoothing story can account for the observed PIs when reducing the dataset size. To that end, we replicate an experiment by Yuan et al. (2020) where the teacher targets are compared to a softened version of the real labels $\delta(y)$ (see the implementation details in the App. D.2.1) across dataset sizes, and report the results in Table 1.

The PI due to label smoothing is almost constant over the dataset size, whereas, as shown before, the PI with distillation is markedly higher at lower dataset sizes (data efficiency). Thus, although when using $100\%$ of the dataset label smoothing and distillation show similar PIs, their behaviour is substantially different for $\kappa < 1$. This confirms that the properties of distillation are not fully captured by label smoothing, which allows us to ultimately reject this hypothesis.

### 3.2 Dark knowledge

Next, we examine the feature learning view of distillation Allen-Zhu & Li (2020), which builds upon the intuition of dark knowledge given by Hinton et al. (2015). The hypothesis is that distillation implicitly pushes the student features to align to the teacher's features. More specifically, let $\phi$ be a non linear feature extractor and $h$ be an affine layer, with $z = h \circ \phi$ being the network's logits. We call $\phi(x)$ the features associated with the input $x$.

We look at the effect of distillation on the student's features, and in particular whether distillation leads to higher feature similarity between the distilled student and the teacher. We study the inner product across the width dimension (which is invariant to permutations of neurons) $k_\phi(x, x') := \langle \phi(x), \phi(x') \rangle$. $k_\phi$ is often referred to as the *feature kernel* (Kornblith et al., 2019). We then compare the feature kernels using the Centered Kernel Alignment (CKA) (Kornblith et al., 2019).

Remarkably, we observe (Figure 2) that the feature kernels of KD students are very aligned to the respective teacher's kernel, consistently across all image classification settings. Comparing Figure 2 to Figure 1 we may hypothesise the existence of a link between the PI and the alignment to the teacher in feature space: both the feature kernel alignment increase and the PI are higher for lower values of $\kappa$. In Figure 17 we find a strong correlation between the two across dataset sizes. Additionally, in App. C.5 we plot the kernel alignment between students trained with different seeds on the same input data, and we observe a significantly higher similarity among the KD students compared to any other pair of trained networks.

To the best of our knowledge *this is the first time logit-based distillation has been observed to result in representational alignment* . The mechanisms giving rise to this phenomenon are not trivial, given that the student only has access to the teacher logits, not features. In App. C.4.4 we begin to investigate in this direction. Although further research is needed to establish whether the different

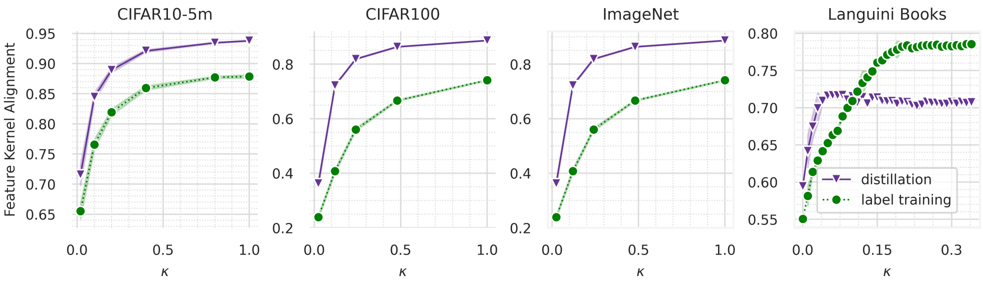

Figure 2: **Distillation induces feature kernel alignment in image classification settings.** On the y-axes the CKA of the feature kernels $k_\phi$ of the KD and LT students to the teacher's feature kernel. Note that the LT students and the LT teacher are both trained with labels. On the x-axis the portion of dataset used. We observe that KD produces markedly steeper curves, yielding high feature kernel alignments at low $\kappa$.

results on language and vision may be reflective of these tasks' different properties, these results indicate that feature learning holds promise for theoretical understanding of distillation.

### 3.3 Student (in)fidelity

Finally, we examine another widely held view on distillation: that with enough data and training, the student should eventually reproduce the teacher (Beyer et al., 2022) (perfect fidelity). Stanton et al. (2021) observed that, perfect fidelity is often neither attainable nor necessary to achieve good performance in practice. However, we are interested in assessing the role of fidelity at lower dataset sizes. In particular, is there a relation between the observed PIs and the degree of fidelity when $\kappa < 1$?

Following Stanton et al. (2021), we measure fidelity using *average Top-1 Agreement* $\mathbb{E}_D[\mathbb{1}\{\text{argmax}_c(p_t(x))_c = \text{argmax}_c(p_s(x))_c\}]$ and focus on self-distillation. Note that fidelity is distinct from feature alignment since it is measured on the outputs of the model, however high feature alignment may be a cause of high fidelity. In contrast to Stanton et al. (2021), we find a strong positive correlation between test fidelity and PI over multiple values of $\kappa$ and across datasets (Figure 3), despite fidelity always falling short of the 100% target. This suggests that alignment with teacher predictions may be a driving factor in the PI on small datasets. Thus, we may revise the conclusions of Stanton et al. (2021) stating that the bulk of the performance increment observed with distillation correlates with the alignment to the teacher, however perfect alignment is not necessary nor achieved in practical settings.

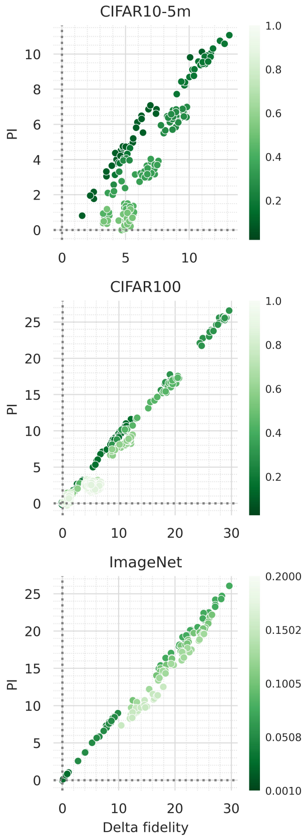

Figure 3: **Fidelity and PI correlate.** Delta fidelity is the difference to the fidelity of an LT student trained on the same amount of data. The colormap represents $\kappa$.

## 4 Discussion and conclusions

In this work we have studied the behaviour of distillation as a function of the dataset size. In so doing we discover a fundamental property of distillation, namely data efficiency. In particular, we find (Figure 1) that the benefits of distillation are significantly enhanced when considering lower fractions of data. Given that the literature on KD so far has focused on a slice of the dataset size axis, we use this novel empirical angle to evaluate existing theories of KD, potentially exposing the empirical bias behind these intuitions. We indeed find that label smoothing is not an accurate model of distillation for $\kappa \neq 1$ and, on the contrary our evidence suggests that feature learning has a central role in KD. This latter finding has further implications on the study of distillation, as currently prominent analyses based on fixed features models cannot capture the complexity of KD in neural networks.

## 5 Related works

We review the relevant literature on distillation in two steps. First, we go over the most popular theories of distillation, and in a second step, we look at existing references to data efficiency in the literature. Generally, we find that the discussion on distillation has mostly focused on its generalisation and knowledge transfer aspects, and that the dataset size has largely been overlooked.

**Distillation: the main threads.** The discussion of generalisation benefits in distillation has produced several different answers, often difficult to reconcile. The oldest and most prevalent account of distillation stems from the intuition that its benefits over hard label targets must reside in the *dark knowledge*, i.e. the class relationship structure implicit in a teacher network's output (Hinton et al., 2015). Allen-Zhu & Li (2020) hypothesise that the students learn independent features from the teacher (due to independent initialisations) and distillation is effective because the teacher is able to transfer otherwise unlearnt features to the student. In contrast to this, Yuan et al. (2020); Zhou et al. (2021) characterise distillation through the lens of *label smoothing regularisation* (Szegedy et al., 2016). According to this view, the better generalisation of distillation is simply a result of the regularising effect of smoothed labels (Müller et al., 2019). Lastly, the findings by Furlanello et al. (2018); Stanton et al. (2021); Nagarajan et al. (2023) further challenge a fundamental intuition on distillation, namely that the student learns a high fidelity representation of the teacher. In particular, Stanton et al. (2021) present compelling evidence that distillation fidelity, defined as the test agreement of the student to the teacher, is typically lower than expected, and higher fidelity does not always imply higher generalisation.

Other works which have experimentally and theoretically contributed to understanding distillation in isolation from the mainstream discussion are (Mobahi et al., 2020; Lopez-Paz et al., 2015; Dong et al., 2019; Beyer et al., 2022; Yim et al., 2017; Zhao et al., 2022).

**Distillation: data efficiency.** Data efficiency in distillation has first been mentioned in the seminal work of Hinton et al. (2015), where the authors briefly examine the regularisation effect of distillation, leading to reduced overfitting in low data regimes. The discussion is however too short to be conclusive. Phuong & Lampert (2019) provide a theoretical model of distillation in a simplified linear binary classification setting. Crucially, they find that distillation benefits from significantly faster statistical convergence rates than those afforded by learning from hard labels , meaning that less data is needed to achieve a given performance when distilling from a trained teacher. This result is improved by Ji & Zhu (2020), who also prove fast statistical convergence rates for distillation in the infinite-width setting. Importantly, both of these results are obtained in fixed-features settings, thus ignoring the potential role of feature learning in the data efficiency of distillation. Other (Foster et al., 2019; Panahi et al., 2022; Zhao & Zhu, 2023; Menon et al., 2021) theoretical accounts of distillation point to data efficiency, however, none of them effectively apply to common practice. Empirical accounts of data efficiency are lacking in the literature. Hao et al. (2024) compare vanilla KD to more sophisticated distillation techniques in high-data regimes. Perhaps the experimental work most similar to ours is Hsieh et al. (2023), who design a *modified distillation* objective *specifically for language models* to obtain data efficiency. Overall, we are the first to comprehensively study distillation data efficiency for practical neural networks and to establish that *distillation dominates label training in low data regimes.*

## Impact Statement

By highlighting data efficiency as a fundamental facet of KD, our study shifts the understanding of how distillation works and opens new pathways for research. This has significant implications for improving model performance in data-scarce environments, which is crucial for fields like medical imaging, autonomous driving, and natural language processing. Our work fosters advancements in deep learning methodologies, promoting more efficient and effective deployment of AI technologies.

## References

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., and Kolesnikov, A. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10925–10934, 2022.

Bordelon, B., Canatar, A., and Pehlevan, C. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pp. 1024–1034. PMLR, 2020.

Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, 2006.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dong, B., Hou, J., Lu, Y., and Zhang, Z. Distillation ≈ early stopping? harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network. *arXiv preprint arXiv:1910.01255*, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Foster, D. J., Greenberg, S., Kale, S., Luo, H., Mohri, M., and Sridharan, K. Hypothesis set stability and generalization. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Hao, Z., Guo, J., Han, K., Hu, H., Xu, C., and Wang, Y. Revisit the power of vanilla knowledge distillation: from small scale to large scale. *Advances in Neural Information Processing Systems*, 36, 2024.

He, B. and Ozay, M. Feature kernel distillation. In *International Conference on Learning Representations*, 2021.

He, B. and Ozay, M. Exploring the gap between collapsed amp; whitened features in self-supervised learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8613–8634. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/he22c.html`.

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

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

Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.-Y., and Pfister, T. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Ji, G. and Zhu, Z. Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. *Advances in Neural Information Processing Systems*, 33:20823–20833, 2020.

Kim, H. and Kim, K. Fixed non-negative orthogonal classifier: Inducing zero-mean neural collapse with feature dimension separation. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=F4bmOrmUwc`.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint 1412.6980*, 2017.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

Menon, A. K., Rawat, A. S., Reddi, S., Kim, S., and Kumar, S. A statistical perspective on distillation. In *International Conference on Machine Learning*, pp. 7632–7642. PMLR, 2021.

Mobahi, H., Farajtabar, M., and Bartlett, P. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Nagarajan, V., Menon, A. K., Bhojanapalli, S., Mobahi, H., and Kumar, S. On student-teacher deviations in distillation: does it pay to disobey? *arXiv preprint arXiv:2301.12923*, 2023.

Nakkiran, P., Neyshabur, B., and Sedghi, H. The deep bootstrap framework: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020.

Panahi, A., Rahbar, A., Bhattacharyya, C., Dubhashi, D., and Haghir Chehreghani, M. Analysis of knowledge transfer in kernel regime. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1615–1624, 2022.

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Phuong, M. and Lampert, C. Towards understanding knowledge distillation. In *International conference on machine learning*, pp. 5142–5151. PMLR, 2019.

Stanić, A., Ashley, D., Serikov, O., Kirsch, L., Faccio, F., Schmidhuber, J., Hofmann, T., and Schlag, I. The languini kitchen: Enabling language modelling research at different scales of compute. *arXiv preprint arXiv:2309.11197*, 2023.

Stanton, S., Izmailov, P., Kirichenko, P., Alemi, A. A., and Wilson, A. G. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919, 2021.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Tang, J., Shivanna, R., Zhao, Z., Lin, D., Singh, A., Chi, E. H., and Jain, S. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*, 2020.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, 2021a.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021b.

Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4133–4141, 2017.

Yuan, L., Tay, F. E., Li, G., Wang, T., and Feng, J. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3903–3911, 2020.

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

Zhang, S., Lyu, Z., and Chen, X. Revisiting knowledge distillation under distribution shift. *arXiv preprint arXiv:2312.16242*, 2023.

Zhao, B., Cui, Q., Song, R., Qiu, Y., and Liang, J. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

Zhao, Q. and Zhu, B. Towards the fundamental limits of knowledge transfer over finite domains. *arXiv preprint arXiv:2310.07838*, 2023.

Zhou, H., Song, L., Chen, J., Zhou, Y., Wang, G., Yuan, J., and Zhang, Q. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. *arXiv preprint arXiv:2102.00650*, 2021.

# Appendices

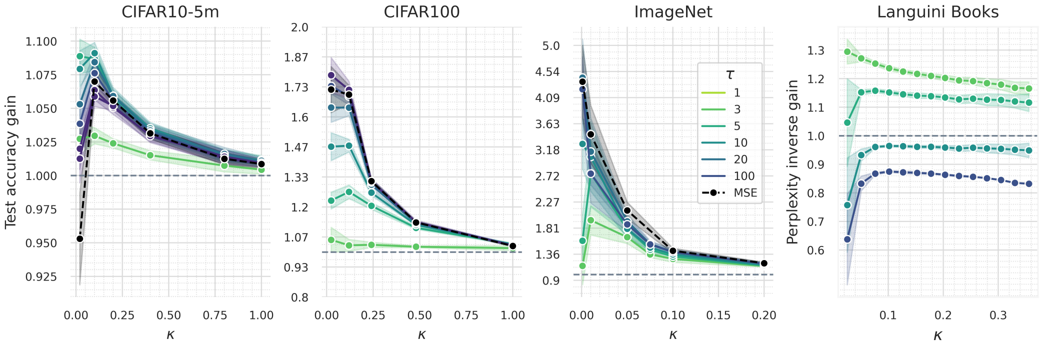

Figure 4: **Effect of temperature.** Comparing the test accuracy gain of distilled students trained at different temperatures. The optimal temperature value varies from task to task. The limit of infinite temperature is represented by the MSE case.

## A Ablation studies

**Metrics** For a given student pair $(p_{KD}, p_{LT})$ we define the *test accuracy gain* of distillation over label training as $Acc(p_{KD})/Acc(p_{LT})$, respectively the *test perplexity inverse gain* as $PPL(p_{KD})/PPL(p_{LT})$ for text data. These gain metrics are chosen such that the larger the values they take, the more the distilled student outperforms the label student.

### A.1 Effect of objective.

We denote the network output distribution by $p_f(x) = \sigma(f(x))$ and the one-hot target distribution by $\delta(y)$. Hereafter, we refer to the following loss as the minimisation objective:

$$(1-\alpha)\cdot \mathbb{E}_{x\sim D}\Big[\ kl\big(\ p_t^\tau(x\ ), p_f^\tau(x)\ \big)\ \Big] + \alpha \cdot \mathbb{E}_{x,y\sim D}\Big[\ kl\big(\ \delta(y),\ p_f(x)\ \big)\ \Big] \tag{1}$$

where $kl(p,q) \coloneqq p^\top \log(p/q)$ is the Kullback-Leibler divergence between distributions $p$ and $q$. When $\alpha = 1$ we recover the cross-entropy loss (*label training*) on the true labels and when $\alpha = 0$ we recover the distillation loss.

Taking inspiration from common practice, we explore three kind of changes to the objective: varying the temperature $\tau$ (Figure 4), the weight $\alpha$ and using soft or hard targets in distillation (Figure 5). Empirically, these aspects of the objective have been shown to affect the generalisation performance of distillation in the classical $\kappa = 1$ scenario.

**Temperature.** We follow the convention by which temperature is applied *symmetrically* to both the teacher and student output distributions. In this case, it is easy to show that temperature scales the gradient by a $\frac{1}{\tau}$ factor and in the limit $\tau \to \infty$ the kl loss gradient converges to the squared loss gradient (cf. Hinton et al., 2015, Eq. 2&4). In other words, we can intuitively think of an increasing temperature as smoothly interpolating from the kl to the squared loss.

**Weight $\alpha$.** In the typical distillation setting ($\kappa = 1$) it is common to use intermediate values of $\alpha$ (Hinton et al., 2015; Tang et al., 2020). We repeat our CIFAR10-5m and CIFAR100 experiments for multiple values of $\alpha$.

**Hard labels.** Finally, Touvron et al. (2021b) have empirically shown that providing the teacher prediction as a true hard label yields lower test errors in some settings. We use hard labels distillation on CIFAR100 and CIFAR10-5m as we vary $\kappa$.

**Results.** The results of our ablations are provided in Figures 4 and 5. In general, we see that what holds true for $\kappa = 1$ does not invariably extend to other dataset sizes, and that different datasets give rise to different behaviours. Importantly, we find that the temperature $\tau$ is crucial to the performance gains and they may even vanish for some temperatures (e.g. see $\tau = 1$ on CIFAR100). Due to the influence of the temperature on the gradients, this evidence seems to suggest that the data efficiency of distillation may be fundamentally understood as a phenomenon concerning the optimisation dynamics. Additionally, using hard teacher targets reduces the performance increments in low data regimes. It

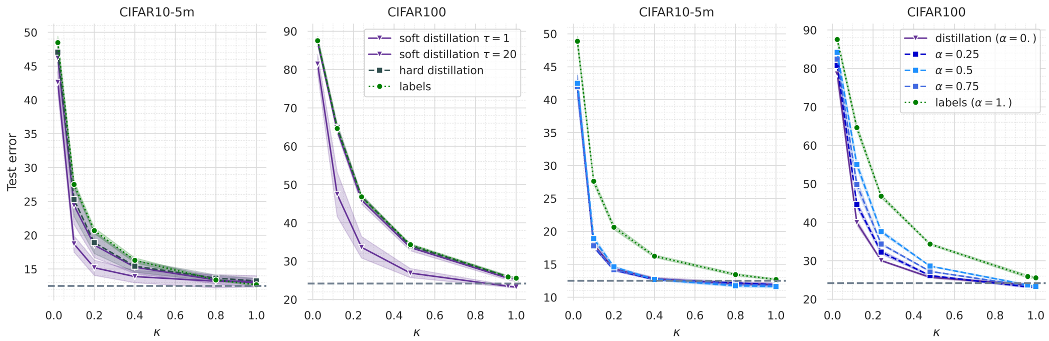

Figure 5: **Effect of soft labels and $\alpha$.** Test error on CIFAR10-5m and CIFAR100 as we (left-half) switch from soft to hard labels distillation and (right-half) vary $\alpha$ . We average over 5 seeds.

follows that the probabilities associated with the non-target classes are necessary for data efficiency, and, as such, they are fundamental to distillation. This observation is in line with the *dark knowledge* hypothesis on distillation, which our analysis so far supported.

## A.2 Effect of model and dataset size

Originally, distillation has been introduced as a technique to reduce the size of a network without sacrificing performance. Therefore it is common in practice to consider teacher and student models of different sizes. Here we play with the relative size of the teacher and student networks as we vary the dataset size, to assess its effect on data efficiency. In particular, we are interested in testing the broad cases of 'teacher *bigger than* the student', 'teacher *smaller than* the student' , and 'teacher *equal to* the student' using two different network sizes. We experiment with different teacher-student combinations on CIFAR10 and Languini Books, summarised in the legend of Figure 6 (as $P_{Teacher} \rightarrow P_{Student}$). Additionally, we extend the range of $\kappa$ beyond 1 (up to 10) in the CIFAR10-5m dataset,in order to gauge the behaviour of distillation in the *high data* regime as well. In App. D.1 we report the experiment configuration used in each plot of the paper.

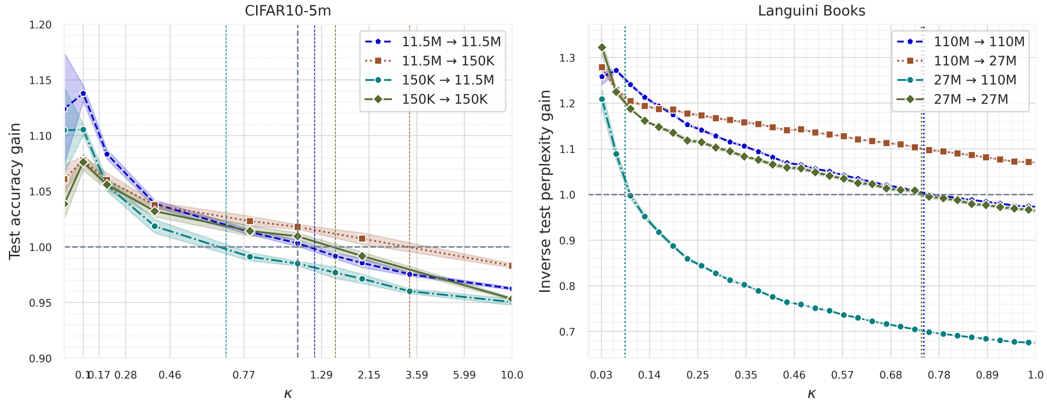

Figure 6: **Effect of relative size.** Depicted is the gain in performance on CIFAR10-5m and Languini Books, averaged over 5 seeds. The vertical dashed lines mark the intersection point $\kappa^{*}$ for each configuration. Observe that the student *relative size* $P_{Student}/P_{Teacher}$ (negatively) correlates with $\kappa^{\star}$.

Firstly, from Figures 1 and 6 we observe that the performance increment diminishes as $\kappa$ increases. Training with ground-truth labels outperforms distillation when using more data than the teacher has been trained on (recall that the teacher is trained with one-hot class targets at $\kappa = 1$), or a more powerful model. Notably, the distilled student's performance flattens out slightly above the teacher's for $\kappa > 1$. This slight increase in performance when $\kappa > 1$ aligns with previous observations in the literature on *self-distillation* (Furlanello et al., 2018; Allen-Zhu & Li, 2020; Stanton et al., 2021).

Moreover, the fact that the error of the distilled student converges to somewhere close to the teacher's error, regardless of the model size, is in line with the bias-variance argument of Menon et al. (2021). Simply, in the high-data regime the non zero bias term penalises distillation over label training. We discuss this view more in depth in App. C.2.

In Figure 6, we mark by a vertical line the value of $\kappa$, such that at values of $\kappa > \kappa^\star$ label training outperforms knowledge distillation. We observe that the *relative size* of the student with respect to the teacher correlates (negatively) with $\kappa^\star$. For each dataset, a higher relative size $P_{Student}/P_{Teacher}$ corresponds to a lower $\kappa^\star$: in CIFAR10 $P_{Student}/P_{Teacher} \approx 76.66$ and $\kappa^\star \approx 0.7$, and in Languini Books $P_{Student}/P_{Teacher} \approx 4.07$ and $\kappa^\star \approx 0.083$. These observations hint at a relationship of the form $\kappa^\star = \beta \cdot (P_{Student}/P_{Teacher})^{-1}$, where $\beta$ is a setting-specific constant. What does this behaviour tell us about the relationship between distillation and label training? Recalling that distillation converges to a performance close to the teacher's we can draw a simple conclusion. The value of $\kappa^\star$ mostly reflects the amount of data needed to reach the teacher performance when training with labels. In the case of self-distillation, this value will be close to 1, and as we increase the overparametrization of the student (with respect to the teacher) $\kappa^\star$ decreases.

## B Can DED be useful in practice?

Table 2: **Distillation with transfer learning.** Validation accuracy for distillation and labels training on a sweep of datasets. The teacher has been pretrained in a supervised fashion on ImageNet1k and adapted to each dataset by retraining only the linear head. The $^\star$ symbol indicates hyperparameter tuning: the teacher head training and student label training use respectively the best hyperparameter discovered by grid search. For comparison, we use the same hyperparameters between distillation and labels training. Nonetheless, distillation outperforms labels training on almost all the benchmarks.

| | FLOWERS | DTD | AIRCRAFT | CALTECH | CARS | FOOD |
|---|---|---|---|---|---|---|
| # TRAINING POINTS | 1020 | 1880 | 6667 | 7810 | 8144 | 75750 |
| # CLASSES | 102 | 47 | 10 | 101 | 196 | 101 |
| DISTILLATION | **41.37** ± 1.60 | **37.04** ± 6.01 | **54.71** ± 1.86 | **73.98** ± 0.82 | **73.84** ± 0.65 | 75.09 ± 0.26 |
| LABELS$^\star$ | 35.84 ± 1.41 | 28.36 ± 2.45 | 53.40 ± 4.30 | 71.64 ± 1.03 | 70.20 ± 1.54 | **81.84** ± 0.34 |
| TEACHER$^\star$ | **86.60** | **67.44** | 45.18 | **94.00** | 55.03 | 70.76 |

We have shown with our work that distillation is data efficient, and it is natural to ask whether any practical advantage could be derived from it for neural network training in realistic settings.

In order to answer this question we first consider distillation in data scarce scenarios, where data efficiency could be beneficial to reach lower test errors. We mimic such scenarios with a transfer experiment (Table 2), where we use a pre-trained teacher available on the PyTorch hub and finetune only its linear head using the available data. Zhang et al. (2023) have studied KD in the case of distribution shifts, however we are specifically interested in small datasets, which is outside of their focus. We repeat the experiment on several publicly available datasets, which vary in size and number of classes. We compare distillation and label training using the entire available data. Notice that, since the teacher has been pre-trained on Imagenet-21k, $\kappa$ is low for almost all the datasets considered. Due to the little data available, hyperparameter tuning makes a significant difference in final performance. Therefore, to avoid biasing our result in favour of distillation, we do hyperparameter tuning on label training and use the same optimal hyperparameters also for distillation. We also tune the hyperparameters when training the teacher linear head on the new data. Astonishingly, distillation outperforms label training in almost all tasks considered, only falling behind when the dataset is effectively large. This remarkable result suggests that distillation can be helpful in applications with severely constrained resources, and we believe that further research into the behaviour of distillation in low data regimes can lead to further boosts in performance.

Next, we move on to assess whether *data efficiency* translates into *computational efficiency* and thus whether distillation may be used to reduce the training costs of realistic neural networks. In Figure 7 we plot the test perplexity of GPT-MINI networks trained with distillation or label training as a function of the *flops*. We compare distillation with teacher of different sizes, while keeping the student the same. Perhaps unsurprisingly, we find that the computational costs incurred by the extra forward pass through the teacher penalise distillation over label training. In particular, as the teacher size is increased the final test perplexity lowers but the computational cost increases. Ultimately, distillation is not computationally efficient in 2 out of 3 scenarios considered.

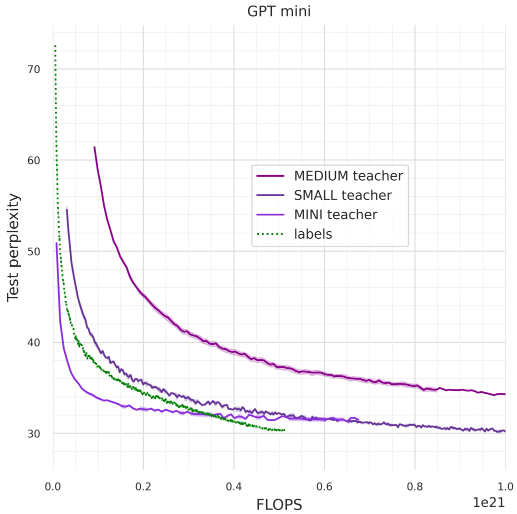

Figure 7: **Distillation is not computationally efficient.** Test perplexity over FLOPS for GPT mini students with 3 different teachers. A teacher of bigger size corresponds to higher PI but also higher FLOPS.

# C Additional Material to the main paper discussion

## C.1 DED in Vision-Transformers.

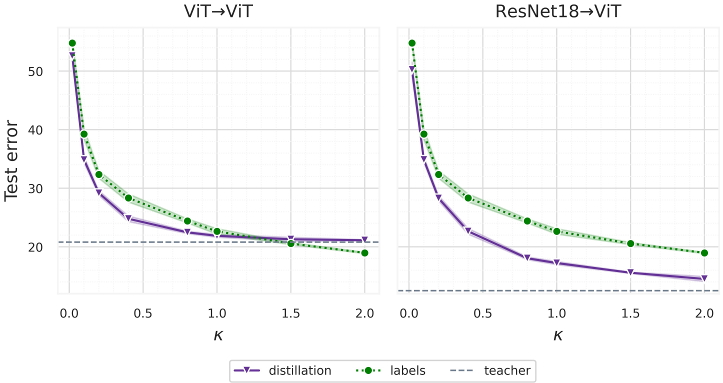

Figure 8: **DED can be observed in attention-based architectures.** Test error on CIFAR10-5m as a function of the relative training dataset size for ViT models. Compared are models obtained through label training and distillation from a ViT teacher (left) and a ResNet18 teacher (right). Importantly, we observe data efficiency also for attention-based architectures when using distillation.

Out of curiosity and completeness in our empirical analysis we run an experiment using Vision Transformers (ViT) on CIFAR10-5m. Given that ViTs are notoriously data inefficient and the CIFAR10 dataset is relatively small, the ViT teacher we use (adapted from this Pytorch implementation of (Dosovitskiy et al., 2020), without extra data augmentations for better comparisons with CNNs) only achieves 80% validation accuracy on CIFAR10. Therefore, we also compare the setting of training ViT students with the ResNet18 teacher. In Figure 8 we plot the test error of distillation and label training as we vary the fraction of training data $\kappa$. Interestingly, the performance increment is consistently higher when using the ResNet18 teacher, and it carries over the $\kappa = 1$ threshold. We suspect that the reason for this difference lies in the markedly lower test error in the ResNet18 teacher, however, further experiments are needed to finalise this claim.

### C.2 Bias-Variance tradeoff.

We turn to a simple bias-variance decomposition of the expected error, in a similar spirit as (Menon et al., 2021). Let $p_s(D)$ be a student trained on the dataset $D$ and $\bar{p}_s^M$ be the mean student trained with $M$ samples, i.e. $\bar{p}_s^M = \mathbb{E}_{D\sim\mathcal{P}^M}[p_s(D)]$. Taking $p_y$ to be the true label distribution[1], the expected squared loss $l_2(f,g) = \mathbb{E}_{x,y}[\|f(x) - g(x)\|^2]$ decomposes into two terms:

$$\mathbb{E}_{D\sim\mathcal{P}^M}[l_2(p_s(D),p_y)] = \underbrace{\mathbb{E}_D[l_2(p_s(D),\bar{p}_s^M)]}_{\text{Variance}} + \underbrace{l_2(\bar{p}_s^M,p_y)}_{\text{Bias}^2} + \epsilon \tag{2}$$

where $\epsilon$ is an irreducible approximation error. As the number of training samples grows $M \to \infty$, the variance term reduces up to the noise inherent in the optimisation process. Consequently, the bias term controls the behaviour in the high-data regime for both distillation and label training. In the case of distillation with a fixed teacher trained on finite data, the bias term converges to a constant, which depends on the teacher accuracy on $\mathcal{P}$, as well as the bias implicit in the optimisation procedure. Thus, in the high-data regime, the positive bias penalises distillation over ground-truth targets. By the same token, when the data is scarce the variance term may be significantly higher than the bias and dominate the error. Therefore the presence of increased PI in low data regimes suggests that distillation has a variance reduction effect on the estimator, which compensates for the higher bias. And this effect is consistent across datasets and models.

### C.3 Zooming into (in)fidelity.

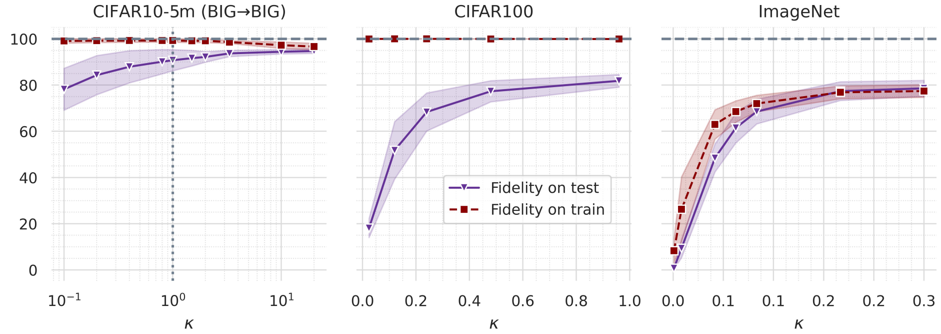

Figure 9: **Fidelity of self distillation in low and high data regimes.** On the y-axes we plot the distillation fidelity (in percent) on train and test data, averaged over 5 seeds and 8 temperature values.

We report additional results on distillation fidelity for the CIFAR10-5m dataset, which allows us to explore the particularly interesting high-data regime. In Figure 10 we plot distillation fidelity on train and test data for different student-teacher network configurations.

We must remark that several aspects of this setting are sub-optimal and do not match the experiments in Stanton et al. (2021), therefore the conclusions must be taken with a grain of salt. To begin with, the training hyperparameters are not optimised and they are especially inadequate for the 'small' networks. Another factor which may be entangled in these results is the presence of augmentations. We adopt the same augmentations for all network configurations, despite the differences in representational capacity. Finally, in some settings, there is an irreducible approximation error due to the mismatch of student and teacher architecture, which may be a confounder to higher fidelity error.

Nevertheless, we observe an interesting trend in the high data regime. The train and test curves converge to the same value as $\kappa$ increases. We speculate that the 'convergence fidelity error' may quantify the implicit bias given by the optimisation procedure.

[1] Note that by using $p_y$ instead of $\delta(y)$ we get rid of potential label noise.

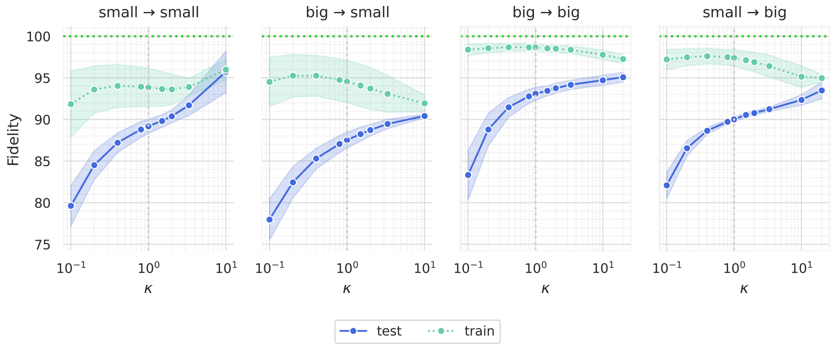

Figure 10: Distillation fidelity over CIFAR10-5m train and test data for different network configurations.

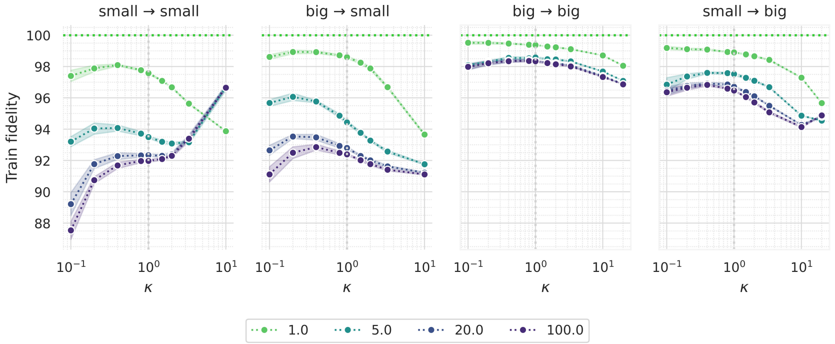

Figure 11: **Temperature affects train fidelity** Distillation fidelity over CIFAR10-5m train data as we vary the distillation temperature $\tau$.

Further, in Figure 11 we show train fidelity for multiple distillation temperatures. Temperature appears to have a strong influence on train fidelity. One hypothesis is that this effect is a consequence of the different training dynamics due to the temperature scaling the gradient. More surprisingly, the trend is reversed with respect to generalisation: higher temperatures deliver higher generalisation and lower train fidelity.

Finally, we plot the difference between train and test fidelity as a function of $\kappa$. Curiously, we find that, across all configurations, the difference curves are well approximated by $O(1/\sqrt{\kappa})$.

### C.4 Distillation and feature learning.

#### C.4.1 What impact does the linear head have on feature learning?

We assess the relevance of the linear *head* $h$ in DED. In other words, we ask: *is the observed data efficiency dependent on the linear map $h$?*

This is a natural question to ask because different feature extractors $\phi$ are known to perform differently when $h$ is trained on little data, depending on the eigendecomposition of $\phi$ (Bordelon et al., 2020; He & Ozay, 2022). To answer this question, we take feature extractors from teacher-distilled and label-trained students, on various fractions $\kappa$ of data, and fit a logistic regression classifier on the feature-based representation *of the whole dataset* ($\kappa$ = 1). By fitting the *linear probe* Alain & Bengio (2016) on the full dataset we are accounting for potential effects of data scarcity on the linear map $h$.

In Figure 13 we show the results. Crucially, we observe that retraining the linear layer *preserves* the gain in test-accuracy of distillation and the effect of temperature (Figure 18) across students, with the

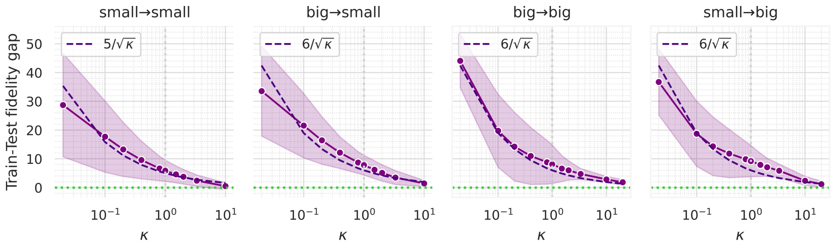

Figure 12: **The difference between train and test fidelity reduces at a $1/\sqrt{\kappa}$ rate.** We plot the difference between train and test fidelity on CIFAR10 for each network configuration. We juxtapose each curve with the best fitting $\omega/\sqrt{\kappa}$ line.

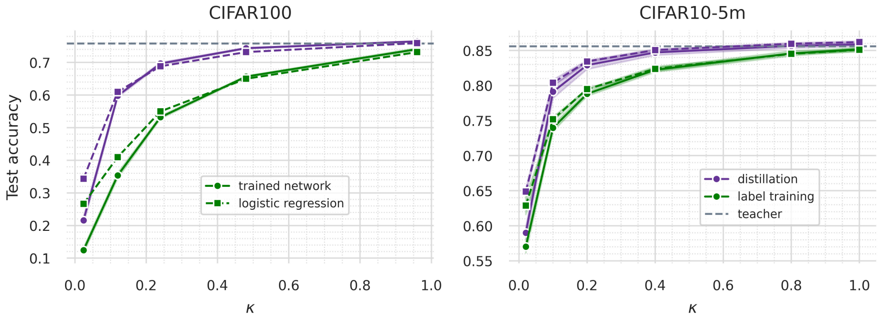

Figure 13: **Data efficiency does not depend on the linear head.** Test classification accuracy (in a 0-1 scale) as a function of $\kappa$. We compare the trained network to a logistic regression classifier (dashed lines). Label-trained students are shown side by side with distilled students (e.g. Figure 14).

largest gains for small $\kappa$ as expected. We therefore conclude that the data efficiency of distillation cannot be captured wholly through the linear layer $h$ and one must consider also the network features.

### C.4.2 Does distillation induce the same features?

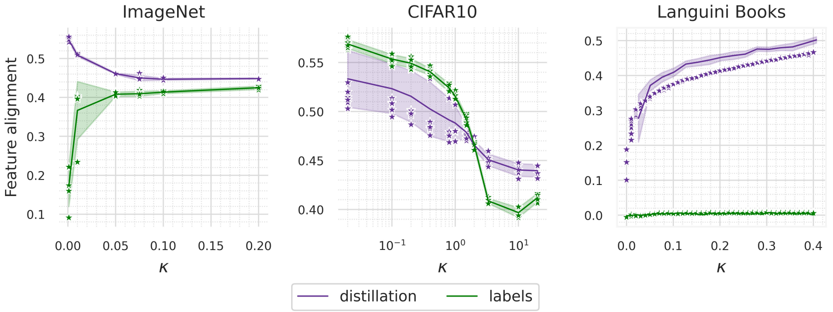

Figure 14: **Feature alignment varies across datasets and architectures.** Feature alignment (Equation (3)) between the teacher and the distillation- (purple) and label- (green) trained students as a function of $\kappa$. The markers show individual samples, and the lines represent the average.

We proceed to explore the effect of distillation on the student network features $\phi$. We ask the following simple question: *do the distillation-trained features approximate the teacher features?*

Table 3: **Feature alignment does not depend on initialisation.** This table reports feature alignment averaged over several values of $\kappa$ for 5 seeds.

| Network | FA | Distillation | Same init |
|---|---|---|---|
| RN18 | $0.49 \pm 0.03$ | √ | × |
| | $0.40 \pm 0.06$ | √ | √ |
| | $0.51 \pm 0.07$ | × | × |
| | $0.52 \pm 0.07$ | × | √ |
| CNN | $0.78 \pm 0.01$ | √ | × |
| | $0.79 \pm 0.01$ | √ | √ |
| | $0.84 \pm 0.01$ | × | × |
| | $0.84 \pm 0.01$ | × | √ |

In order to answer this question we look at the normalised inner product between the students and teacher features when the two networks are identical. More precisely, let $a, b$ be two different instances of the same network, we define their *feature alignment* to be:

$$\mathrm{FA}(a, b) = \frac{1}{Z} \langle \phi_a, \phi_b \rangle_D \tag{3}$$

The sign $\langle \cdot, \cdot \rangle_D$ denotes the average over the data distribution, which we approximate by an average over the test set, and $Z = \sqrt{\langle \phi_a, \phi_a \rangle_D \cdot \langle \phi_b, \phi_b \rangle_D}$ normalises the score.

Figure 14 shows the feature alignment between the students and the teacher on 3 benchmarks of different difficulty. Importantly, feature alignment can only be computed if the teacher and student features are of the same dimension. Thus we apply this test only to the self-distillation settings. We do not observe a shared trend among the benchmarks, suggesting that distillation does not necessarily imply feature alignment. Note that for convolutional networks the features are taken after ReLU activation and thus the alignment will be positive. This is not the case in the transformer network. Perhaps surprisingly, we observe low alignment also when the student and teacher initialisation coincide (Table 3).

### C.4.3 NTK alignment

It is natural to inquire whether the alignment observed at the feature layer propagates back through the network backbone. In order to do this we look at the Neural Tangent Kernel (NTK) (Jacot et al., 2018), a model of training dynamics in wide NNs that is exact in the infinite-width limit under certain parameterisations. In the NTK setting, an NN $f_\theta$ evolves as a linear model in its parameters $\theta$, with a *fixed* feature map determined by its Jacobian $\frac{\partial f_\theta}{\partial \theta}$ at initialisation, which captures features from all layers in the NN.

Importantly, the (last layer) feature kernel appears in the NTK computation as one summand in a sum over the network layers, because the Jacobian of $f_\theta$ with respect to the last linear layer is precisely the feature vector $\phi$. Therefore the NTK alignment between two networks captures offers an overview of the alignment of the feature at all the intermediate layers.

We compute the NTK of teacher and student networks (both distillation and labels) and evaluate their alignments using CKA. We plot the result in Figure 15, alongside the feature-kernel alignments for the same experimental setting. Predictably, we observe a similar trend in the two curves. However, the feature-kernel alignment is generally higher than the NTK's, suggesting that the effect of distillation is best observed in the feature layer.

### C.4.4 Does distillation yield feature kernel alignment?

First, the CKA is defined as follows:

$$\mathrm{CKA}(k_s, k_t) = \frac{\mathrm{HSIC}(k_s, k_t)}{\sqrt{\mathrm{HSIC}(k_s, k_s) \cdot \mathrm{HSIC}(k_t, k_t)}} \tag{4}$$

with $\mathrm{HSIC}(k_s, k_t) = (n-1)^{-2} \cdot \mathrm{Tr}(k_s H k_t H)$, and $H$ being a centering matrix.

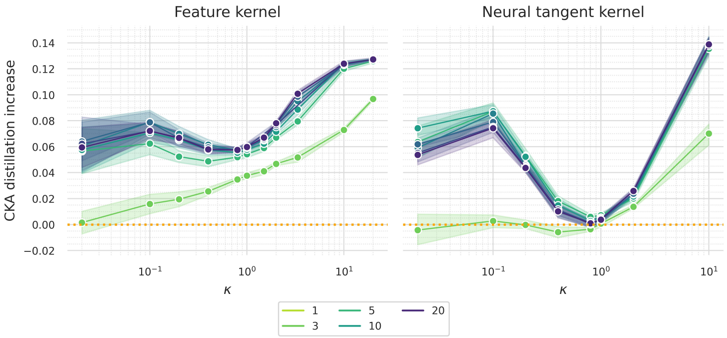

Figure 15: **NTK vs FK alignment.** The kernels are measured on CIFAR10 for the SMALL→SMALL network configuration.

We begin by looking at the case of an optimal distillation student $f_s^\star$. Say that $f_s^\star(x) = f_t(x)$ for all $x \in D_M$, ($D_M$ being the training dataset of size $M$). If we define the *target kernel* as:

$$k_f^{tg}(x, x') \coloneqq \langle f(x), f(x') \rangle \tag{5}$$

it is obvious to conclude that distillation entails equivalence of the teacher and student's target kernels on the training data (cf Tang et al. (2020) for evidence of this effect). However, it is not obvious how feature kernel alignment may ensue from target kernel alignment. Rewriting $f_s^\star(x)$ as $W_s\phi_s(x)$ and $f_t(x)$ as $W_t\phi_t(x)$ the target kernel is $k_f^{tg}(x, x') = \phi_s(x)^\top [W_s^\top W_s]\,\phi_s(x')$. Thus from the equivalence of target kernels, it follows that:

$$\phi_s(x)^\top [W_s^\top W_s]\,\phi_s(x') = \phi_t(x)^\top [W_t^\top W_t]\,\phi_t(x')$$

If $W_s$ and $W_t$ are orthogonal matrices, we can immediately conclude that the student and teacher feature kernels are equivalent up to some scaling factor.

But in general $W_s$ and $W_t$, will not be square matrices and cannot be orthogonal. Indeed, for image classification settings we will have the output projection down to a smaller number of classes than width, and for language modelling transformers we have the opposite (the Languini vocabulary is 16k). For the image classification setting, we can hope to recover some structure in the feature and weight spaces due to the Neural Collapse phenomnon Papyan et al. (2020); Kim & Kim (2024), which will tells us that the features and the weights in trained classification NNs on small numbers of classes will become aligned. They will also exhibit a Simplex Equiangular Tight Frame behaviour in the final layer, where class inputs are mapped to the class centroid. Investigating if Neural Collapse can help to explain the feature alignment we observe with distllation provides an interesting direction for future work.

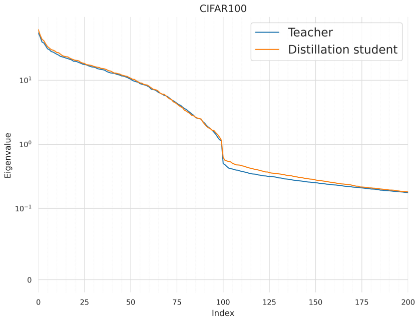

Figure 16: Eigenspectrum for teacher and a distillation student network trained on CIFAR100. We observe a drop after the first 100 dimensions, which is often indicative of neural collapse.

### C.5 Extra plots

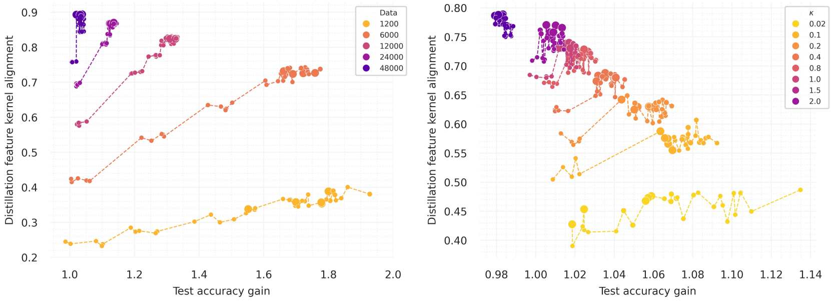

Figure 17: **Feature kernel alignment correlates with test accuracy gain.** Each point represents a different student-pair instance for varying $\kappa$ (represented by the colour) and $\tau$ (represented by the size) on CIFAR100 (left) and CIFAR10 (right). The dashed lines connect points with the same $\kappa$ to highlight the differences within equivalent data regime groups.

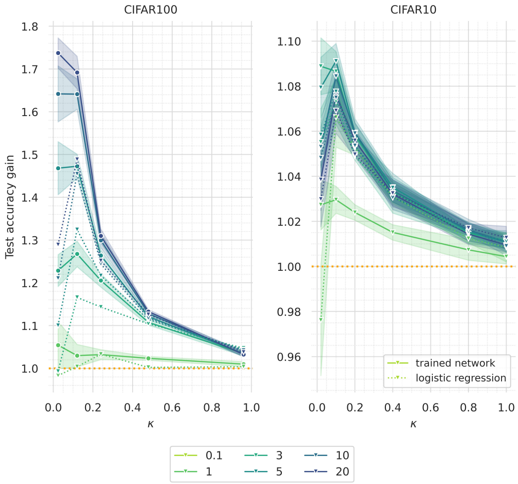

Figure 18: **Data efficiency does not depend on the linear head (2).** Test accuracy gain as a function of $\kappa$ and the distillation temperature $\tau$. We compare the trained network to a logistic regression classifier.

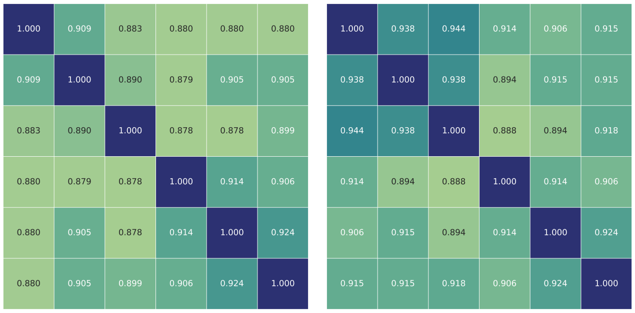

Figure 19: $\kappa = 0.02, \tau = 1$ (left) and $\tau = 20$ (right).

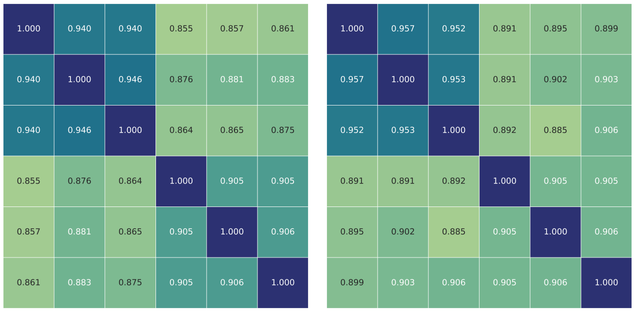

Figure 20: $\kappa = 0.1, \tau = 1$ (left) and $\tau = 20$ (right).

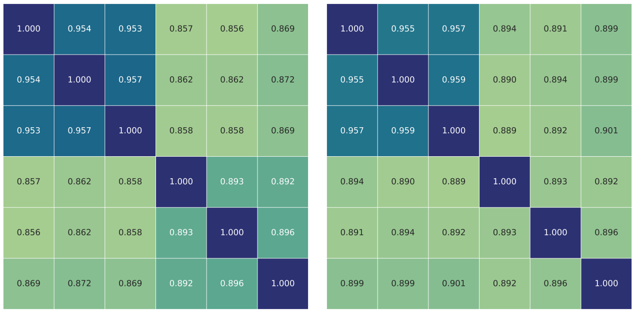

Figure 21: $\kappa = 0.2, \tau = 1$ (left) and $\tau = 20$ (right).

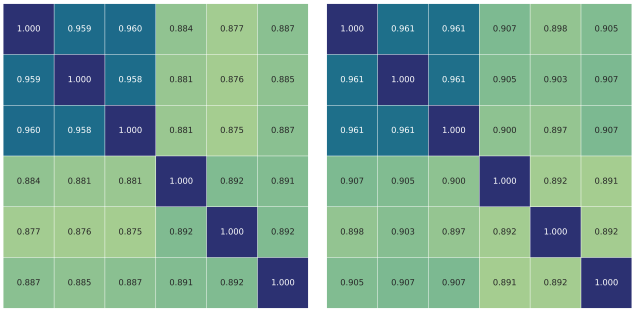

Figure 22: $\kappa = 0.2, \tau = 1$ (left) and $\tau = 20$ (right).

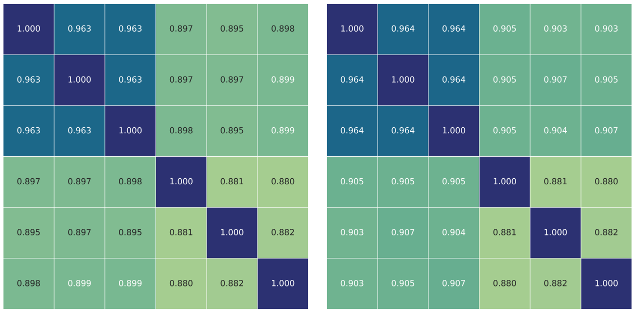

Figure 23: $\kappa = 0.4, \tau = 1$ (left) and $\tau = 20$ (right).

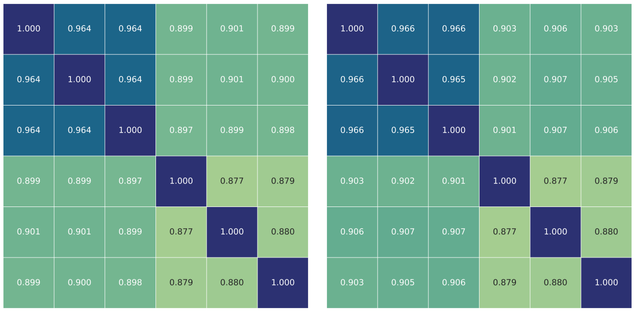

Figure 24: $\kappa = 1.0, \tau = 1$ (left) and $\tau = 20$ (right).

# D Details on the experimental setup

## D.1 Dataset, Networks & Configurations

We repeat our experiments on 4 different datasets, namely CIFAR10-5m (C10) (Nakkiran et al., 2020), CIFAR100 (C100) (Krizhevsky & Hinton, 2009), IMAGENET (IMN) (Deng et al., 2009) and *Languini Books (LBOOKS)* (Stanić et al., 2023) , and several networks. In particular, for the image datasets we use a set of convolutional networks and for the LBOOKS dataset we use GPT networks of varying sizes. An overview of the experiments configuration is given in Table 4. We use a publicly available extended version of CIFAR10 figuring around 6 mln images, synthetically generated by sampling from a generative model trained on CIFAR10 (commonly named CIFAR 5m). We evaluate our models on the test set also included in the CIFAR 5m collection. The dataset has been released together with the paper (Nakkiran et al., 2020).

Table 4: **Overview of the experiments configurations.** The lines marked by the $^\star$ symbol refer to experiments presented in the Appendix.

| Dataset | Student Networks ($P$) | Teacher Networks | SELF | Name |
|---|---|---|---|---|
| CIFAR10 (+5m) | Vanilla CNN (150K) | Vanilla CNN (150K) | √ | SMALL→SMALL |
| | | ResNet18 (11.5M) | × | BIG→SMALL |
| | ResNet18 (11.5M) | Vanilla CNN (150K) | × | SMALL→BIG |
| | | ResNet18 (11.5M) | √ | BIG→BIG |
| | ViT (6.3M)$^\star$ | ViT (6.3M) | √ | - |
| | | ResNet18 (11.5M) | × | - |
| CIFAR100 | ResNet18 (11.5M) | ResNet18 (11.5M) | √ | - |
| IMAGENET | ResNet50 ( 25.6M) | ResNet50 (25.6M) | √ | - |
| LANGUINI BOOKS | GPT MINI (27M) | GPT MINI (27M) | √ | MINI→MINI |
| | GPT MINI (27M) | GPT SMALL (110M) | × | SMALL→MINI |
| | GPT MINI (27M) | GPT MEDIUM (336M) | × | MEDIUM→MINI |
| | GPT MINI2 (67M) | GPT MEDIUM (336M) | × | MEDIUM→MINI2 |
| | GPT SMALL (110M) | GPT MINI (27M) | × | MINI→SMALL |
| | GPT SMALL (110M) | GPT SMALL (110M) | √ | SMALL→SMALL |

**Exact configuration in each plot and table.** For the CIFAR10 and Languini Books dataset we report the network configuration used in each plot shown in the main paper:

- Figure 1 C10: BIG→BIG, LBOOKS: SMALL→SMALL.
- Figure 4 C10: SMALL→SMALL, LBOOKS: MEDIUM→MINI2.
- Figure 5 C10: BIG→BIG
- Figure 6 C10: all except those including ViT, LBOOKS: MINI→MINI, SMALL→MINI, MINI→SMALL, SMALL→SMALL.
- Table 1 C10: BIG→SMALL
- Figure 9 C10: BIG→BIG
- Equation (4) C10: SMALL→SMALL, LBOOKS: MEDIUM→MINI2.
- Figure 24 C10: SMALL→SMALL

### D.1.1 Range of $\kappa$.

Exact set of values of $\kappa$ used for each dataset:

- C10: $[0.02, 0.1, 0.2, 0.4, 0.8, 1., 1.5, 2., 3.3, 10.20.]$
- C100: $[0.024, 0.12, 0.24, 0.48, 1.0]$
- IMN: $[0.001, 0.01, 0.05, 0.075, 0.1, 0.2, 0.3]$

- LBOOKS: We train GPT-like language models on the Languini Books dataset in a streaming fashion, i.e. each batch is processed only once. Therefore, $\kappa$ dynamically increases during training.

CIFAR10-5m is a synthetic dataset of similar distribution as CIFAR10 with ~ 6M instead of 60K samples. This allows us to investigate $\kappa \gg 1$ for teachers pre-trained on CIFAR10, as discussed in Section 2. In particular, we perform experiments using up to 20× more data than the teacher training data with CIFAR10-5m.

#### D.1.2 Network architectures

In line with common practice, all our networks are of the form, $f(x) = (h \circ \phi)(x)$, for non-linear feature extractor $\phi$ and linear map $h$. Hereafter we may refer to $\phi$ as the network *backbone* and to $h$ as the network *head*. Unless stated otherwise, all the head layers take the form of a linear map from the feature space $\phi$ to the logit space $z$: $h(\zeta) = W\zeta + b$, $W$ being the weight matrix and $b$ the bias.

**Vanilla CNN** The backbone consists in a sequence of $4$ convolutional layers interleaved by *batch normalisation* layers, *ReLU* activations and *max-pooling*. After flattening, the feature layer has width $160$.

**ResNets** We reproduce the original structure of residual convolutional networks described by He et al. (2016). We use a *ResNet18* (feature layer width $512$) for CIFAR10 and CIFAR100, and a *ResNet50* (feature layer width $1024$) for ImageNet.

**GPT** We use the GPT2-inspired transformer model provided in the Languini benchmark (Stanić et al., 2023). In our experiments we employ $4$ GPT2 models of different sizes. In particular, the width and depth (measured in number of *attention blocks*) of the backbone changes between sizes, but all the models share the same block type. The code of the Languini library is publicly available on GitHub[2]. The *MINI* GPT network has width $512$ and depth $4$; the *MINI2* GPT network has width $1024$ and depth $4$; the *SMALL* GPT network has width $768$ and depth $6$; and finally the *MEDIUM* GPT network has width $1024$ and depth $24$. We use two trained MINI and MEDIUM networks as teachers.

### D.2 Training procedures

All our experiments involve two training steps. First, we train one teacher network on the full dataset (or a fixed portion thereof in case of C10 and LBOOKS data). Second, we train another network (the student) on a variable portion of the dataset.

**Teachers** We train 1 teacher for C100 and IMN,2 teachers for C10 and 3 teachers for LBOOKS. The seed of the teacher is fixed and once trained we use the teacher as a black-box function. Importantly, the teachers are trained with one-hot-labels following common practices (see App. D.2.1 for details).

The C100 and IMN teachers are trained on the full training set. The C10 teachers are trained on a fixed random sample of 60K images from the almost 6M available samples. To ease comparison, the LBOOKS teachers are trained on the same amount ($\approx 8.3G$) of tokens.

**Students** For each experimental configuration, we train two identical networks (which we call students) *with identical training settings*, either using one-hot-labels or soft-label targets provided by the teacher. Each experiment is repeated over $5$ seeds, which means a total of $10$ networks (with $5$ different initialisations). For each dataset, we train these $10$ networks on multiple fractions of data (identified by the value $\kappa$, see App. D.1.1 above). Moreover, we distil all students with different temperatures $\tau$ (see App. D.2.1 for the list).

Notice that a student trained with one-hot labels on the full dataset ($\kappa = 1$) is equivalent to the teacher (up to its initialisation). For this reason, we keep the same training setup for teachers and students. Moreover, we do not change training hyperparameters between label training and teacher distillation to allow for a better comparison.

[2] https://github.com/languini-kitchen

### D.2.1 Hyperparameters

We repeat all of our experiments over $5$ seeds, which affect the network initialisation and the data sampling processes. Moreover, we vary the temperature of distillation in the range $[0.1, 1, 3, 5, 10, 20, 100]$, and we simulate the case $\tau \rightarrow \infty$ with an $l_2$ loss on the logits (cf (Hinton et al., 2015)). Finally, unless stated otherwise, we use the SGD optimiser for training.

For C10 we do not use optimal training hyperparameters. Therefore, the performance achieved by teacher and student networks is not maximal with respect to their capacity. For all the other datasets, however, we rely on publicly available optimal "training recipes" which have been tuned to the architecture. Therefore in the case of C100, IMN and LBOOKS the performance of our models is high relative to the model capacity.

**CIFAR10** For both the teacher and the student networks pair we use the following training hyperparameters: learning rate = $0.1$, with a linear warmup over the first $5$ epochs and subsequently annealing the learning rate with a cosine schedule, weight decay = $0.001$, batch size $256$, $30$ epochs. We use random augmentations consisting of crops to $32 \times 32$ and horizontal flips.

**CIFAR100** For both the teacher and the student networks pair we use the following training hyperparameters: learning rate = $0.1$, with a linear warmup over the first epoch and subsequently reducing the learning rate by a factor of $5$ after $60, 120$ and $160$ epochs, weight decay = $0.0005$, momentum = $0.9$, batch size $128$, $200$ epochs. We use random augmentations consisting of crops to $32 \times 32$, horizontal flips and rotations of $15$ degrees maximum.

**IMAGENET** For both the teacher and the student networks pair we use the following training hyperparameters: learning rate = $0.1$, reducing the learning rate by a factor of $10$ every $30$ epochs, weight decay = $0.001$, momentum = $0.9$, batch size $64$, $90$ epochs. We use random augmentations consisting of crops to $224 \times 224$ and horizontal flips.

**LANGUINI BOOKS** For each GPT model we follow the standard training recipe provided by the Languini library, including Adam Kingma & Ba (2017) (cf the code for details). Importantly, we decay the learning rate at every step and always use a batch size of $128$. The *MINI* teacher has been trained on $3.2$B tokens and the *MEDIUM* teacher has been trained on $5.7$B tokens from the same source.

**Label smoothing** In our label smoothing experiments on C100 we use the same hyperparameters as Yuan et al. (2020) for better comparison (although they use a different student-teacher network configuration). We then repeat the experiment on C10 (this dataset is not present in Yuan et al. (2020)) using the same hyperparameters. Specifically, we set $a = 0.99$ and $\alpha = 0.9$ (so the distillation weight is $0.1$). Moreover, we explore 3 temperature values, namely $\tau = 1, 20, 100$.

### D.2.2 Compute resources

We perform all of our experiments on graphic cards NVIDIA 4090, with 24GB of GPU memory. For the larger language experiments which require higher GPU memory we parallelise our experiments over multiple devices. The maximal runtime of a single experiment is 5 days and 22 hours. The total recorded compute for the entire project (so including failed and omitted experiments) is 1080 days.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide evidence for data efficiency in distillation in Figure 1, we study the factors affecting data efficiency in Section 2 and we apply the data efficiency perspective to validate existing theories of KD in Section 3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We evaluate our experimental setup on several datasets, discussing the contradictory findings between language and vision in Section 3.2. We fully explore the effect of implicit assumptions on our finding such as model size (Figure 6), and other hyperparameters in Section 2.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We report all the information needed to reproduce our experiments in App. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We do not provide access to code in the reviewing phase but we pledge to release the code upon acceptance. Additionally our experiments are mostly based on popular benchmarks in the literature, using common architectures and publicly available datasets, with the exception of Languini Books.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all such details in App. D, and include an overview of the experimental setting in the main paper (**??**).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Every Figure and Table in the paper reports the mean and standard deviation (represented by the shaded area in the plots) over 5 runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report high level information regarding the compute resources in App. D.2.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the code of ethicss and we have not detected any violations.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include an impact statement after the conclusion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not introduce new datasets or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We include citations for all datasets and models used. We do not employ further assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.