Progressive distillation improves feature learning via implicit curriculum

Anonymous Authors[1]

Abstract

Knowledge distillation, where a student model learns from a teacher model, is a widely-adopted approach to improve the training of small models. A known challenge in distillation is that a large teacher-student performance gap can hurt the effectiveness of distillation, which prior works have aimed to mitigate by providing intermediate supervision. In this work, we study a popular approach called *progressive distillation*, where several intermediate checkpoints of the teacher are used successively to supervise the student as it learns. Using sparse parity as a testbed, we show empirically and theoretically that these intermediate checkpoints constitute an implicit curriculum that accelerates student learning. This curriculum provides explicit supervision to learn underlying features used in the task, and, importantly, a fully trained teacher does not provide this supervision.

1. Introduction

Knowledge distillation enables compression of a large, capable *teacher* model into a small *student* model. A plethora of works across different tasks and domains have demonstrated that distillation is an effective learning algorithm, but there is little understanding of when and how distillation is better than learning from ground-truth labels. Prior work has suggested that teachers provide richer information (Lopez-Paz et al., 2016; Tang et al., 2020; Menon et al., 2021; Dao et al., 2021) or better regularization (Yuan et al., 2020; Mobahi et al., 2020; Nagarajan et al., 2024). However, there is also evidence that a large gap in capabilities between the teacher and the student can negatively impact the success of distillation (Cho & Hariharan, 2019; Mirzadeh et al., 2019). One commonly suggested fix is to use additional supervision to bring the student and teacher behaviors closer to one another (Mirzadeh et al., 2019; Jin et al., 2019; Jafari et al., 2021; Harutyunyan et al., 2022).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

This work focuses on a particular instantiation of this idea, which we call *progressive distillation*, where the student receives supervision from intermediate checkpoints of the teacher.[1] Progressive distillation has grown increasingly popular in practice (Anil et al., 2018; Jin et al., 2019; Harutyunyan et al., 2022), and is thought to improve the generalization of the student by modulating task difficulty to follow the student's capability during training (Harutyunyan et al., 2022; Jafari et al., 2021). However, in this work, we demonstrate instead that the benefits of progressive distillation can be better characterized through how it improves the *optimization* of the student model.

We use the classical sparse parity task (O'Donnell, 2014; Edelman et al., 2023; Abbe et al., 2024) as our testbed, where the input is a vector of ± 1 values and the label is given by the parity of some unknown subset of the coordinates (i.e., the *support*). It is well known that for sparse parity on n bits with a size-k support, the amount of computation required by SQ learning is $\Omega(n^k)$ (Kearns, 1998). In this setting, we demonstrate that progressive distillation enables the student to learn with less data and fewer optimization steps than what is required when learning from the data alone, circumventing the SQ lower bound (Section 3). Specifically, in Section 4, we show that progressive distillation provides an implicit curriculum, as intermediate checkpoints of the teacher reveals information about the support of the sparse parity. In addition to empirical evidence, we show formally that such implicit curriculum reduces the number of online SGD steps required by the student compared to either learning from ground-truth labels or distilling only from a fully trained teacher (Theorem 4.1). Our findings shed light on when and how distillation provides a benefit over learning directly from the data.

Related works. One persistent surprise in knowledge distillation is that increasing the strength of the teacher does not necessarily lead to improved student performance. Prior works have speculated that an overly large "teacher-student gap" may make it difficult for the student to follow the teacher and thus proposed to bridge this gap by introducing supervision of intermediate difficulty (Mirzadeh et al., 2019; Cho & Hariharan, 2019; Harutyunyan et al., 2022; Jafari

[1]Several works also refer to progressive distillation as online distillation (Anil et al., 2018; Harutyunyan et al., 2022).

et al., 2021). Mirzadeh et al. (2019) adopted a multi-step distillation strategy using models of intermediate sizes, and Shi et al. (2021) proposed a technique to directly inject teacher supervision into the student's trajectory using an approximation of mirror descent. Theory on what types of labels provide the strongest learning signal motivated using a moving average of the teacher to supervise the student (Ren et al., 2022). Most similar to our work, Harutyunyan et al. (2022) analyzed distillation for extremely wide networks and found it helpful to learn from the intermediate checkpoints of the teacher. They speculated that this is because neural networks learn progressively complex functions during training (Kalimeris et al., 2019). In contrast to their work characterizing the generalization ability of the student, we study the optimization dynamics of distillation.

2. Setup

We study the **sparse parity** task, which is commonly used as a testbed in understanding optimization (Barak et al., 2022; Bhattamishra et al., 2022; Morwani et al., 2023; Edelman et al., 2023; Abbe et al., 2024). The input $\mathbf{x}$ is a boolean vector picked uniformly at random from the n-dimensional hypercube $\{\pm 1\}^n$. The label $y \in \{\pm 1\}$ is determined by some size-d subset S of the n coordinates. In particular, $y = \prod_{i \in S} \mathbf{x}_i$. It is well-known that the sparse parity problem is difficult to learn and requires $\Omega(n^d)$ samples when using online SGD (Barak et al., 2022; Edelman et al., 2023).

Extension to hierarchical parity: We also consider a hierarchical extension of the sparse parity task. In this setting, labels are assigned using a binary tree, where the edge taken from each node is determined by the parity of the coordinates within that node. The hierarchical nature of the task requires the model to learn many features, each of which can require different sample complexity, so it provides further insight into how progressive distillation may behave on more complex problems. We defer an extensive discussion of this setting to Appendix C but mention relevant results alongside the standard parity setting.

2.1. Distillation strategies

Let $f_{\mathcal{T}}, f_{\mathcal{S}}$ denote the teacher and the student models, respectively, each of which outputs real-valued logits over C classes. The logits are turned into a probability distribution using $\text{softmax}(f_{\mathcal{S}}/\tau)$ for some temperature hyperparameter τ. We always use $\tau = 1$ for the student (Zheng & Yang, 2024), and default to $\tau = 1$ in the teacher unless otherwise specified. Given a teacher model $f_{\mathcal{T}}$, the distillation loss for the student $f_{\mathcal{S}}$ on a sample x with label y is defined as

$$L_\alpha(x, y; f_{\mathcal{S}}, f_{\mathcal{T}}) = \alpha L_{\text{CE}}(\text{softmax}(f_{\mathcal{S}}(x)), y) + (1-\alpha) L_{\text{KL}}(\text{softmax}(f_{\mathcal{S}}(x)), \text{softmax}(f_{\mathcal{T}}(x)/\tau)), \quad (1)$$

where L_{CE} is the cross entropy loss, $L_{\text{KL}}(f_{\mathcal{S}}, f_{\mathcal{T}}) := -\sum_{i \in [C]} f_{\mathcal{T}} \log f_{\mathcal{S}}$ is the KL loss for distillation, and $\alpha \in [0, 1]$ is a hyperparameter for weighting ground-truth supervision against teacher supervision. Our experiments set $\alpha = 0$ to isolate the effect of teacher supervision.

We consider two strategies for choosing the teacher. The first is *one-shot distillation*, where $f_{\mathcal{T}}$ is fixed throughout training to the last-iterate checkpoint. The second strategy is *progressive distillation*, where the student learns from (multiple) intermediate checkpoints of a teacher's training run, denoted by $\{f_{\mathcal{T}}^{(1)}, \cdots, f_{\mathcal{T}}^{(N)}\}$ for some N. There are many ways to choose these $\{f_{\mathcal{T}}^{(i)}\}$. A generically applicable strategy is to choose $\{f_{\mathcal{T}}^{(i)}\}$ at some fixed intervals in the teacher's training run (Anil et al., 2018; Harutyunyan et al., 2022). There is often a trade off in choosing the interval: too frequent checkpointing makes optimization easier, but requires more storage for the checkpoints. Interestingly, we find that a few (or even one) checkpoints suffice to drastically speed up the training of the student (Section 3).

3. Progressive Distillation Accelerates Training

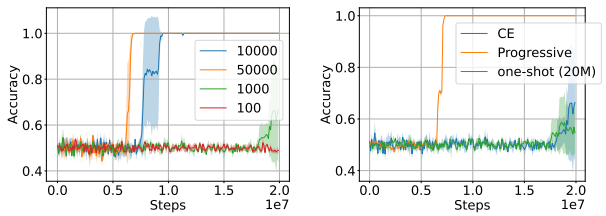
This section empirically highlights the benefit of progressive distillation. We compare the following training strategies:

1. *Cross-entropy (CE) training* (i.e. Eq. 1 with $\alpha = 1$),
2. *One-shot distillation* from the teacher's final checkpoint ($\alpha = 0$).
3. *Progressive distillation* from teacher's checkpoints in regular intervals ($\alpha = 0$).

Experiment Details. The teacher and student models are 1-hidden-layer MLPs with ReLU activation. The teacher has a hidden width of 5×10^4, and the students are of widths 10^2 or 10^3. All models are trained using SGD with batch size 1 for $20M$ steps on sparse parity data with $n = 100$ and $d = 6$ (Section 2). The learning rate is searched over $\{10^{-2}, 5 \times 10^{-3}, 10^{-3}\}$. Evaluation is based on a held-out set consisting of 4096 examples, and we report the average across 3 different training seeds. For strategy (2) we use the teacher checkpoint at the end of training (20M checkpoint), and for strategy (3) we take checkpoints that are 0.5M steps apart.

Small models require longer training to learn sparse parities. We observe that when using CE training, wider models learn the sparse parity much faster (Figure 1a), consistent with findings in prior work (Edelman et al., 2023). Among distillation strategies, with the default temperature 1, the student barely benefits from distilling the final teacher checkpoint (Figure 1b). In contrast, progressive distillation allows the student to learn at the same speed as a much wider teacher and reach a perfect accuracy.[2]

[2]More results are shown in Figure 4.



(a) (b)

Figure 1. (a) **Wider models learn faster** from ground truth labels. We show accuracy curves for models of different widths trained on sparse parity data with $n = 100$ and $\mathbb{S} = 6$ (Section 2). (b) **Progressive distillation at regular intervals accelerates learning in a student with width 1000**. Similar observations hold for width-100 students (Figure 4) and hierarchical data (Figures 3 and 5).

How many checkpoints do we need? From a practical standpoint, it is desirable to use fewer checkpoints in progressive distillation as checkpoints can be expensive to store and load. As such, we also test another strategy that we call *p-shot Distillation*, which is progressive distillation with $p-1$ intermediate checkpoints and the final checkpoint. For sparse parity, using as few as 1 intermediate checkpoint (i.e. $p = 2$) suffices to significantly accelerate training, as shown in (Figure 2, right). We find it most useful to use a checkpoint taken during the "phase transition" of the accuracy (Figure 2, left), a choice we will justify in the next section.

4. Mechanistic Understanding: Progressive Distillation Provides Implicit Curriculum

In this section, we demonstrate that the intermediate checkpoints constitute an implicit curriculum that accelerates student learning when performing progressive distillation.

4.1. Monomial Curriculum

Learning sparse parity with noisy gradients is a type of learning with statistical queries (SQ), for which the $O(n^d)$ SQ lower bound applies. When learning with neural networks, Edelman et al. (2023) showed that the neurons can be viewed as parallel queries. Therefore, treating the product of network width and the number of training steps as proportional to the number of queries, the SQ lower bound implies a fundamental trade-off between the width and the steps, where narrower networks require more steps to learn.

However, as this section will show, distillation can help circumvent such lower bound by providing an implicit curriculum. To see why, note that learning sparse parity with neural networks generally requires two steps: searching for the support S, for which a large width is required, and subsequently computing the product of variables in the support, i.e. $\prod_{i\in S} \mathbf{x}_i$. Distillation is helpful because a wide teacher can first learns the support S, and provide the student with outputs highly correlated[3] with degree-1 monomials in the support, i.e. $\mathbf{x}_i, \forall i \in S$. Learning from this monomial reduces the sample complexity required for the student model to learn the support. We note that this is a specific instantiation of the curriculum that neural networks are broadly known to undertake when learning sparse parity (Barak et al., 2022; Edelman et al., 2023). Below, we demonstrate that the teacher empirically obeys this curriculum (Section 4.2), followed by a theoretical justification (Section 4.3).

4.2. Empirical Evidence for the Monomial Curriculum

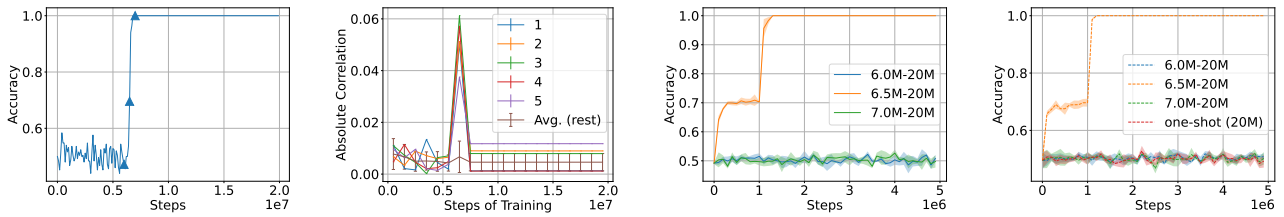
We demonstrate that at certain intermediate checkpoints, the teacher's logits correlate strongly with the aforementioned support monomials (Figure 2b). This correlation diminishes as training proceeds, and the final teacher checkpoint provides little signal as to what the support is. Notably, the correlation spikes at the time step where the teacher's accuracy dramatically increases. Furthermore, we observe that for 2-shot distillation, only the teacher's checkpoint *during phase transition* helps train a student to 100% accuracy (Figure 2). **For hierarchical data**, the correlations with variables in different features can emerge at different time steps (Figure 8). As such, we observe that 2-shot distillation fails to train a model to 100% accuracy, and 3-shot distillation with checkpoints selected based on emergence of different features can instead help train a student model to 100% accuracy (Figure 7). Our findings suggest that more complex tasks likely require more intermediate checkpoints.

The succes of progressive distillation does not come from soft label regularization. One potential hypothesis for the benefit of intermediate checkpoints is that earlier checkpoints provide "softer" (as opposed to one-hot) labels, which prior works suggest have a regularization benefit (Yuan et al., 2020). Intuitively, softer labels allow the student to have smaller weight norms, hence acting as weight regularization. However, we show that this hypothesis does *not* explain progressive distillation: We repeat the experiments in Figure 2 using hard labels from the teacher (by setting the temperature $\tau = 10^{-4}$), thereby removing any potential regularization effects induced by soft labels. Our findings are largely unchanged, suggesting that it is the monomial curriculum, not regularization, that is the key mechanism to the success of progressive distillation (Figure 2d).

4.3. Theoretical justification

We now formalize the benefits of progressive distillation for the d-sparse parity problem. The student $f_{\mathcal{S}}$ and the teacher $f_{\mathcal{T}}$ models are 1-hidden layer MLPs with ReLU activations, whose sizes are determined by the hidden layer

[3] Correlation to a polynomial g is measured by $\mathbb{E}_{\mathbf{x}} g(\mathbf{x}) f_{\mathcal{T}}(\mathbf{x})$



(a) Teacher accuracy behavior

(b) Correlations $\mathbb{E}_{\mathbf{x},y} f_{\mathcal{T}}(\mathbf{x}, y)x_i$ for $i \in S$ during training

(c) 2-shot Distillation ($\tau = 1$)

(d) 2-shot Distillation ($\tau = 10^{-4}$)

Figure 2. (a) Teacher exhibits a sharp phase transition in accuracy between $6M$ and $7M$ steps during training. 3 candidate choices of intermediate checkpoints for 2-shot Distillation are marked by triangles ($\{6M, 6.5M, 7M\}$ checkpoints). (b) During the phase transition, the teacher's output $f_{\mathcal{T}}$ shows higher correlation to monomials of variables in the support, compared to monomials not in the support. (c) Teacher's checkpoint during phase transition ($6.5M$) helps 2-shot Distillation performance to converge to 100% accuracy, while other checkpoints don't. (d) Even with extremely low temperature, the benefit of the phase transition checkpoint persists, suggesting that the monomial curriculum, not soft label regularization, is the key to the success of progressive distillation. For the 2-shot Distillation in (c, d), the student is trained with an intermediate checkpoint for $1M$ steps, followed by distillation from the final teacher checkpoint until the end of training. Student is of width 1000. Results for a student of width 100 are in Appendix Figure 6.

width. Following previous works (Barak et al., 2022; Edelman et al., 2023), we analyze a simplified two-stage training procedure and modify the loss function to use the hinge loss: $L_\alpha(\mathbf{x}, y; f_{\mathcal{S}}, f_{\mathcal{T}}) = \alpha \max(0, 1 - f_{\mathcal{S}}(\mathbf{x})y) + (1 - \alpha)\max(0, 1 - f_{\mathcal{S}}(\mathbf{x})f_{\mathcal{T}}(\mathbf{x}))$. This modification allows us to verify the existence of the monomial curriculum by computing the correlation of the student's output to either the true label or the teacher's output.

When training from true labels, there is a gap in the teacher's weights between coordinates in and out of the support S. We show that the magnitude of the correlations between $f_{\mathcal{T}}$ and the monomials $x_i, \forall i \in S$, is at least $\Omega(1/d)$ at this stage (Theorem B.5). As training progresses, the correlations to the monomials diminish (Theorem B.7).

Recall that learning S requires $\Omega(n^d)$ samples under supervision of true labels (Edelman et al., 2023). In contrast, under supervision of the form $\sum_{i\in S} \mathbf{x}_i + g$ where g is a higher order polynomial over S, the model can learn the support S with $\Theta(n)$ samples. However, as the correlation to the support monomials decreases, the necessary sample complexity moves toward learning from true labels only.

This comparison can be formalized by the sample complexity gaps between one-shot distillation from a $\mathcal{O}(n^{-c})$-error teacher model $f_{\mathcal{T}}$ (for $c \geq 3$) and progressive distillation. For progressive distillation, we assume that we have access to the teacher checkpoint that has correlations of magnitude $\Omega(1/d)$ to the monomials $\mathbf{x}_i, \forall i \in S$. We train the student model on that checkpoint for a few steps, and then switch to distilling from the $\mathcal{O}(n^{-c})$-error checkpoint, similar to what is done in the experiments. Formally, we show:

Theorem 4.1 (Informal version of Theorem B.8)**.** *Consider learning d-sparse parity with a student model of size $\tilde{m} \geq \tilde{\Omega}(2^d)$. Suppose, we are learning from a teacher with loss $\mathcal{O}(n^{-c})$ error for some $c \geq 3$. Then, the total sample complexity needed for the student to reach $\mathcal{O}(\epsilon)$-loss for progressive distillation is $\tilde{\Theta}(2^d n^2 \epsilon^{-2} + d^2)$. However, one-shot distillation requires at least $\Omega(n^{\min(2c,d)})$ samples.*

Remark 4.2. One difference in the analysis compared to the experiments is that the former uses SGD with large batch sizes. A potential direction to bridge this gap is to use analyses similar to those in Abbe et al. (2023), which studies online SGD with Gaussian data. Moreover, our experiments use SGD with fresh samples and batch size 1, as the sample complexity lower bounds also imply lower bounds on the number of optimization steps. Understanding lower bounds on sample complexity and optimization steps for settings like minibatch SGD and multi-epoch training (Dandi et al., 2024) is another interesting future direction.

5. Conclusion

This work studies how knowledge distillation affects the optimization of the student, with a focus on feature learning in classification. Motivated by the teacher-student gap, we study *progressive distillation* methods where the student learns from intermediate checkpoints of the teacher, as opposed to the standard one-shot distillation where the student can only access supervision from one (typically fully-trained) teacher. We find that progressive distillation benefits the learning of the student by providing an *implicit curriculum*, complementing prior literature that identified the benefits of distillation in terms of generalization and regularization. Specifically, using sparse parity and its variants as testbeds, we show theoretically and empirically that the intermediate teacher checkpoints provide supervision that can accelerate student learning. We leave it to future work to extend our experiments and analysis to different tasks.

References

Abbe, E., Boix-Adserà, E., and Misiakiewicz, T. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. *Annual Conference Computational Learning Theory*, 2023. doi: 10.48550/arXiv.2302.11055.

Abbe, E., Cornacchia, E., and Lotfi, A. Provable advantage of curriculum learning on parity targets with mixed inputs. *Advances in Neural Information Processing Systems*, 36, 2024.

Anil, R., Pereyra, G., Passos, A., Ormándi, R., Dahl, G. E., and Hinton, G. E. Large scale distributed neural network training through online distillation. *International Conference on Learning Representations*, 2018.

Barak, B., Edelman, B. L., Goel, S., Kakade, S. M., Malach, E., and Zhang, C. Hidden progress in deep learning: SGD learns parities near the computational limit. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/884baf65392170763b27c914087bde01-Abstract-Conference.html.

Bhattamishra, S., Patel, A., Kanade, V., and Blunsom, P. Simplicity bias in transformers and their ability to learn sparse boolean functions. *Annual Meeting of the Association for Computational Linguistics*, 2022. doi: 10.48550/arXiv.2211.12316.

Cho, J. H. and Hariharan, B. On the efficacy of knowledge distillation. *arXiv preprint arXiv: 1910.01348*, 2019.

Dandi, Y., Troiani, E., Arnaboldi, L., Pesce, L., Zdeborová, L., and Krzakala, F. The benefits of reusing batches for gradient descent in two-layer networks: Breaking the curse of information and leap exponents. *arXiv preprint arXiv: 2402.03220*, 2024.

Dao, T., Kamath, G. M., Syrgkanis, V., and Mackey, L. Knowledge distillation as semiparametric inference, 2021.

Edelman, B. L., Goel, S., Kakade, S., Malach, E., and Zhang, C. Pareto frontiers in neural feature learning: Data, compute, width, and luck. *arXiv preprint arXiv:2309.03800*, 2023.

Harutyunyan, H., Rawat, A. S., Menon, A. K., Kim, S., and Kumar, S. Supervision complexity and its role in knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2022.

Jafari, A., Rezagholizadeh, M., Sharma, P., and Ghodsi, A. Annealing knowledge distillation. *Conference of the European Chapter of the Association for Computational Linguistics*, 2021. doi: 10.18653/v1/2021.eacl-main.212.

Jin, X., Peng, B., Wu, Y., Liu, Y., Liu, J., Liang, D., Yan, J., and Hu, X. Knowledge distillation via route constrained optimization. *IEEE International Conference on Computer Vision*, 2019. doi: 10.1109/ICCV.2019.00143.

Kalimeris, D., Kaplun, G., Nakkiran, P., Edelman, B. L., Yang, T., Barak, B., and Zhang, H. SGD on neural networks learns functions of increasing complexity. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3491–3501, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/b432f34c5a997c8e7c806a895ecc5e25-Abstract.html.

Kearns, M. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. Unifying distillation and privileged information, 2016.

Menon, A. K., Rawat, A. S., Reddi, S., Kim, S., and Kumar, S. A statistical perspective on distillation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7632–7642. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/menon21a.html.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. *AAAI Conference on Artificial Intelligence*, 2019. doi: 10.1609/AAAI.V34I04.5963.

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

Morwani, D., Edelman, B. L., Oncescu, C.-A., Zhao, R., and Kakade, S. Feature emergence via margin maximization: case studies in algebraic tasks. *arXiv preprint arXiv: 2311.07568*, 2023.

Nagarajan, V., Menon, A. K., Bhojanapalli, S., Mobahi, H., and Kumar, S. On student-teacher deviations in distillation: does it pay to disobey?, 2024.

O'Donnell, R. *Analysis of boolean functions*. Cambridge University Press, 2014.

Ren, Y., Guo, S., and Sutherland, D. J. Better supervisory signals by observing learning paths. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Iog0djAdbHj.

Shi, W., Song, Y., Zhou, H., Li, B., and Li, L. Follow your path: a progressive method for knowledge distillation, 2021.

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.

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.

Zheng, K. and Yang, E.-H. Knowledge distillation based on transformed teacher matching. *arXiv preprint arXiv:2402.11148*, 2024.

A. Additional background

Leap complexity The leap complexity is a notion provided by Abbe et al. (2023) that quantifies the difficulty of learning hierarchical structure. As noted in Section 4, for boolean functions, [4] the leap complexity roughly corresponds to the size of the growth in support. We now provide the formal definition in Abbe et al. (2023). Given a boolean function $h : \{\pm 1\}^n \to \{\pm 1\}$, write h in the Fourier basis as

$$h(z) = \sum_{S \in \{0,1\}^n} \hat{h}(S)\chi_S(z), \tag{2}$$

where $\hat{h}(S) := \langle h, \chi_S \rangle$ denote the Fourier coefficients, and $\chi_S(z) := \prod_{i \in S} z_i^{S_i}$.

Given this decomposition, the leap complexity is defined to be the maximum growth in support (i.e. S) at each step, with the optimal ordering of the polynomials in the decomposition. Formally:

Definition A.1 (Leap complexity (Abbe et al., 2023))**.** Given a boolean function h, let $\mathcal{S}(h) := \{S_1, \cdots, S_l\}$ denote the set of non-zero basis elements of h, for some $l \in \mathbb{Z}_+$, an $S_j \in \{0,1\}^n$. The leap complexity of h is defined as

$$\text{Leap}(h) := \min_{\pi \in \prod_l} \max_{i \in [l]} \|S_{\pi(i)} \setminus \cup_{j=0}^{i-1} S_{\pi(j)}\|_1, \tag{3}$$

where $\|S_{\pi(i)} \setminus \cup_{j=0}^{i-1} S_{\pi(j)}\|_1 := \sum_{k \in [n]} S_{\pi(i)}(k)\mathbb{I}\{S_{\pi(j)}(k) = 0, \forall j \in [i-1]\}$, with $S_{\pi(0)} = 0^n$.

B. Formalization of Section 4.3

The teacher model is defined as

$$f_{\mathcal{T}}(\mathbf{x}) = \sum_{i=1}^{m} a_i \sigma\left(\langle \boldsymbol{w}_i, \mathbf{x} \rangle + b_i\right).$$

The student model is similarly defined as

$$f_{\mathcal{S}}(\mathbf{x}) = \sum_{i=1}^{\tilde{m}} \tilde{a}_i \sigma\left(\langle \tilde{\boldsymbol{w}}_i, \mathbf{x} \rangle + \tilde{b}_i\right).$$

Following Abbe et al. (2023) and Barak et al. (2022), we adopt a two-stage batch gradient descent training, where we first train the first-layer weights $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m\}$, keeping the output weights $\{a_i\}_{i=1}^m$ fixed . In the second stage of training, we fit the output weights $\{a_i\}_{i=1}^m$ while keeping others fixed. We keep the biases $\{b_i\}_{i=1}^m$ fixed throughout training. Similar strategy for training the student model as well. The teacher is trained with hinge loss, given by $L(\mathbf{x}, y) = \max(0, 1 - f_{\mathcal{T}}(\mathbf{x})y)$. The student is trained with $L_\alpha(\mathbf{x}, y; f_{\mathcal{S}}, f_{\mathcal{T}}) = \alpha \max(0, 1 - f_{\mathcal{S}}(\mathbf{x})y) + (1-\alpha)\max(0, 1 - f_{\mathcal{S}}(\mathbf{x})f_{\mathcal{T}}(\mathbf{x}))$.

Data: We assume the data points are sampled at random from $\mathcal{U}(\{\pm 1\}^n)$. W.l.o.g., let the target d-sparse parity function be $y = x_1 x_2 \cdots x_d$.

Notations

- S denotes the support of the sparse parity.
- At any training step t, $f_{\mathcal{T}}^{(t)}$ will refer to the teacher's output at that step. Its parameters are referred to as $\theta^{(t)} = \{a_i^{(t)}, \boldsymbol{w}_i^{(t)}, b_i^{(t)}\}_{i=1}^m$. The loss for $f_{\mathcal{T}}^{(t)}$ is denoted by $L_{\theta^{(t)}}$. Notations for the student $f_{\mathcal{S}}$ are defined similarly.
- Maj : $\{\pm 1\}^n \to \pm 1$ represents the majority function on n-dimensional boolean data. On any $\mathbf{x}$, Maj returns the sign of $\sum_{i=1}^n \mathbf{x}_i$. ζ_i for $i \geq 1$ represents its kth ourier coefficient, i.e. $\zeta_i = \mathbb{E}_{\mathbf{x},y}\text{Maj}(\mathbf{x})\chi_S(\mathbf{x})$ for any $S \in \{0,1\}^n$ with $|S| = i$. $\zeta_i = 0$ when i is even, and $\zeta_i = \Theta(i^{-1/3}/\binom{n}{i})$ when i is odd (O'Donnell, 2014).
- τ_g denotes the error tolerance in the gradient estimate due to mini-batch gradient estimation: let g be the population gradient and $\hat{g}$ be the estimated gradient with a few examples, τ_g is defined such that $\|\hat{g} - g\|_\infty \leq \tau_g$. A τ_g-error gradient estimate can be obtained using a batch size of $\tilde{O}(1/\tau_g^2)$.

[4] The leap complexity can be defined for any function in L^2. For the purpose of this paper, we provide a definition for the special case of boolean functions only.

Symmetric Initialization: Following Barak et al. (2022), we use the following symmetric initialization: for each $1 \le i \le m/2$,

$$\boldsymbol{w}_i \sim \mathcal{U}(\{\pm 1\}^n), \quad b_i \sim \mathcal{U}(\{-1+d^{-1}, \cdots, 1-d^{-1}\}), \quad a_i \sim \mathcal{U}(\{\pm 1/m\}),$$
$$\boldsymbol{w}_{i+m/2} = \boldsymbol{w}_i, \quad b_{i+m/2} = b_i, \quad a_{i+m/2} = -a_i.$$

Algorithm 1 2-stage training

Require: Stage lengths: T_1, T_2, learning rates η_1, η_2, batch size B_1, B_2, weight decay λ_1, λ_2.
for $t \in [0, T_1]$ and all $i \in [m]$ **do**
Sample B_1-samples $\{(\mathbf{x}^{(j)}, y^{(j)})\}_{j=1}^{B_1}$.
Update the weights $\boldsymbol{w}_i$ as $\boldsymbol{w}_i^{(t)} \leftarrow \boldsymbol{w}_i^{(t-1)} - \eta_1 \mathbb{E}_{(\mathbf{x},y)\in\{(\mathbf{x}^{(j)},y^{(j)})\}_{j=1}^{B_1}} \nabla_{\boldsymbol{w}_i} \left(L_{\theta^{(t)}}(\mathbf{x}, y) + \lambda_1 \|\boldsymbol{w}_i\|^2 \right)$.
end for
for $t \in [0, T_2]$ and all $i \in [m]$ **do**
Sample B_2-samples $\{(\mathbf{x}^{(j)}, y^{(j)})\}_{j=1}^{B_2}$.
Update the outer layer weights a_i as $a_i^{(t+T_1)} \leftarrow a_i^{(t+T_1-1)} - \eta_2 \mathbb{E}_{(\mathbf{x},y)\in\{(\mathbf{x}^{(i)},y^{(i)})\}_{j=1}^{B_2}} \nabla_{a_i} \left(L_{\theta^{(t+T_1-1)}}(\mathbf{x}, y) + \lambda_2 a_i^2 \right)$.
end for

B.1. Lower bound on sample complexity

We first show that the necessary computation (i.e. the product of network width, number of steps, and number of samples) to learn d-parity for a finite size model is $\Omega(n^d)$. We take the following result from (Edelman et al., 2023):

Theorem B.1 (Width-optimization trade-off, cf. Proposition 3 in Edelman et al. (2023))**.** *For $\delta > 0$, gradient noise $\tau_g > 0$, and model width $m > 0$, if $T \le \frac{1}{2}\binom{n}{d}\frac{\delta\tau_g^2}{m}$, then there exists a (n, d)-sparse parity such that w.p. at least $1-\delta$ over the randomness of initialization and samples, the loss is lower bounded as $L(f_{\mathcal{T}}^{(t)}) \ge 1 - \tau_g$ for all $t \in \{1 \cdots T\}$.*

Hence, for a fixed batch size (and hence a fixed τ_g), we either use a bigger width, or more number of gradient steps (which translates to sample complexity since we are using fresh samples each batch).

B.2. First stage analysis for the teacher

First, we show that with an appropriate learning rate, the magnitude of the weights w_{ij} on coordinates $i \in S$ increases to $\frac{1}{2d}$, while the coordinates $i \notin S$ stay $\mathcal{O}\left(\frac{1}{dn}\right)$ small.

Theorem B.2 (Single step gradient descent, Adapted from Claims 1, 2 in Barak et al. (2022))**.** *Fix $\tau_g, \delta > 0$. Set T_1 as 1. Suppose the batch size $B_1 \ge \Omega(\tau_g^{-2}\log(mn/\delta))$. For learning rate $\eta_1 = \frac{m}{d|\zeta_{d-1}|}$ and $\lambda_1 = 1$, the following conditions hold true for all neurons $i \in [m]$ at the end of first stage of training w.p. at least $1-\delta$.*

1. $\left| w_{ij}^{(1)} - \frac{\operatorname{sign}(a_i^{(0)}\zeta_{d-1})\operatorname{sign}(\chi_{[d]\setminus\{j\}}(\boldsymbol{w}_i^{(0)}))}{2d} \right| \le \frac{\tau_g}{|\zeta_{d-1}|}$, *for all $j \in [d]$.*
2. $\left| w_{ij}^{(1)} - \frac{\zeta_{d+1}}{|\zeta_{d-1}|}\frac{\operatorname{sign}(a_i^{(0)})\operatorname{sign}(\chi_{[d]\cup\{j\}}(\boldsymbol{w}_i^{(0)}))}{2d} \right| \le \frac{\tau_g}{|d\zeta_{d-1}|}$, *for all $j > d$.*

Proof. The proof is given in Barak et al. (2022), which we outline here for completeness. The proof has two major components: First, the magnitude of the population gradient at initialization reveals the support of the sparse parity. Second, the batch gradient and the population gradient can be made sufficiently close given a sufficiently large batch size. We will explain each step below.

Claim B.3. *At initialization, the population gradient of the weight vector in neuron i is given by*

$$\mathbb{E}_{\mathbf{x},y} \nabla_{w_{ij}} f_{\mathcal{T}}^{(0)}(\mathbf{x}, y) = -\frac{1}{2} a_i^{(0)} \zeta_{d-1} \chi_{[d]\setminus\{j\}}(\boldsymbol{w}^{(0)}), \quad \text{for all } j \in S$$
$$\mathbb{E}_{\mathbf{x},y} \nabla_{w_{ij}} f_{\mathcal{T}}^{(0)}(\mathbf{x}, y) = -\frac{1}{2} a_i^{(0)} \zeta_{d+1} \chi_{[d]\cup\{j\}}(\boldsymbol{w}^{(0)}), \quad \text{for all } j \notin S$$

Thus, the gradient of the weight coordinates w_{ij} for any neuron i and $j \in S$ has magnitude $|\zeta_{d-1}|$, while the gradients of the weight coordinates w_{ij} for any neuron i and $j \notin S$ has magnitude $|\zeta_{d+1}|$. The gap between the gradient in support and out of support is given by $|\zeta_{d-1}| - |\zeta_{d+1}| \geq 0.03((n-1)^{-(d-1)/2})$ (Lemma 2 in (Barak et al., 2022)).

The second component involves applying a hoeffding's inequality to show the gap between sample and population gradient.

Claim B.4. *Fix $\delta, \tau_g > 0$. For all i, j, for a randomly sampled batch of size B_1, $\{(\mathbf{x}_k, y_k)\}_{k=1}^{B_1}$, with probability at least $1-\delta$,*

$$\left|\mathbb{E}_{\mathbf{x},y\sim\mathcal{U}(\{\pm\}^n)} \nabla_{w_{ij}} f_{\mathcal{T}}^{(0)}(\mathbf{x}, y) - \mathbb{E}_{\{(\mathbf{x}_k,y_k)\}_{k=1}^{B_1}} \nabla_{w_{ij}} f_{\mathcal{T}}^{(0)}(\mathbf{x}, y)\right| \leq \tau_g,$$

provided $B_1 \geq \Omega(\tau_g^{-2} \log(mn/\delta))$.

Because we want the noise τ_g to be smaller than the magnitude of the true gradients for the coordinates in the support S, we want τ_g to be smaller than $|\zeta_{d-1}|$. We set this to get favorable condition for second phase of training (see Theorem B.6). □

On the other hand, we show that after the first phase, the output of the network has positive correlations to the individual variables in the support of the label function, and thus the checkpoint after the first phase can be used to speed up training of future models.

Lemma B.5. *Under the event that the conditions in Theorem B.2 are satisfied by each neuron, which occurs with probability at least $1-\delta$ w.r.t. the randomness of initialization and sampling, the output of the model after the first phase satisfies the following correlations:*

1. $\mathbb{E}_{\mathbf{x},y} f_{\mathcal{T}}^{(1)}(\mathbf{x}) x_i \geq \frac{1}{8d} + \mathcal{O}(\tau_g n \, |\zeta_{d-1}|^{-1}) + \mathcal{O}(m^{-1/2})$ *for all $i \in S$.*
2. $\mathbb{E}_{\mathbf{x},y} f_{\mathcal{T}}^{(1)}(\mathbf{x}) x_i \leq \mathcal{O}((dn)^{-1})$ *for all $i \notin S$.*

Proof. Consider a neuron $i \in [m/2]$ and its symmetric counterpart $i + m/2$. W.L.O.G., we assume $\text{sign}(w_{ij}^{(0)}) = \text{sign}(a_i^{(0)} \zeta_{d-1})$ for all $j \in [d]$, and $\text{sign}(a_i^{(0)}) = 1$. Recall that d is assumed to be even, hence $\chi_d(\boldsymbol{w}_i^{(0)}) = 1$. Then, the condition in Theorem B.2 can be simplified as

$$w_{ij}^{(1)} = \frac{1}{2d} + v_{ij}, \quad w_{i+m/2,j} = -\frac{1}{2d} - v_{ij}, \text{ for all } j \in [d],$$
$$w_{ij}^{(1)} = \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\text{sign}(w_{ij}^{(0)}) + v_{ij}, \quad w_{i+m/2,j} = -v_{ij}, \text{ for all } j \geq d,$$

where v_{ij} satisfies the following conditions.

$$|v_{ij}| \leq \frac{\tau_g}{|\zeta_{d-1}|}, \text{ for all } j \in [d],$$
$$|v_{ij}| \leq \frac{\tau_g}{|\zeta_{d-1}|}, \text{ for all } j \geq d.$$

Then, the sum of the output of the neurons i and $i + m/2$ on an input $\mathbf{x}$ (ignoring the magnitude of a_i) is given by

$$(f_{\mathcal{T}}^{(1)})_i(\mathbf{x}, y) = \sigma(\frac{1}{2d}\sum_{j=1}^{d} x_j + \langle \boldsymbol{v}_i, \mathbf{x}\rangle + b_i) - \sigma(-\frac{1}{2d}\sum_{j=1}^{d} x_j + \langle \boldsymbol{v}_i, \mathbf{x}\rangle + b_i).$$

In support correlations: We are interested in the correlation of this function to a variable x_u for $u \in S$. We argue for $u = 1$, as the similar argument applies for others. Thus, we are interested in

$$\mathbb{E}_{\mathbf{x},y}(f_{\mathcal{T}}^{(1)})_i(\mathbf{x}, y) x_1 = \mathbb{E}_{\mathbf{x},y}\sigma\left(\frac{1}{2d}\sum_{j=1}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n} \text{sign}(w_{ij}^{(0)}) x_j + \langle \boldsymbol{v}_i, \mathbf{x}\rangle + b_i\right) x_1$$
$$- \sigma\left(-\frac{1}{2d}\sum_{j=1}^{d} x_j - \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n} \text{sign}(w_{ij}^{(0)}) x_j + \langle \boldsymbol{v}_i, \mathbf{x}\rangle + b_i\right) x_1. \tag{4}$$

We focus on the first term; argument for the second term is similar. First of all, we can ignore $\langle \boldsymbol{v}_i, \mathbf{x} \rangle$ incurring an error of $\mathcal{O}(\tau_g n \left|\zeta_{d-1}\right|^{-1})$.

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},y}\sigma\left(\frac{1}{2d}\sum_{j=1}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + b_i\right)x_1 \\
&\quad= \mathbb{E}_{\mathbf{x},y:x_1=+1}\sigma\left(\frac{1}{2d} + \frac{1}{2d}\sum_{j=2}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + b_i\right) \\
&\quad- \mathbb{E}_{\mathbf{x},y:x_1=-1}\sigma\left(-\frac{1}{2d} + \frac{1}{2d}\sum_{j=2}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + b_i\right) \\
&\quad\geq \frac{1}{2d}\mathbb{E}_{\mathbf{x},y}\mathbb{I}\left(\frac{1}{2d}\sum_{j=2}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + b_i \geq 0\right).
\end{aligned}$$

The final step follows from the observation that the argument of σ in the first term is $\frac{1}{d}$ higher than the argument of σ in the second term. This implies that when the first term is non-zero, it's at least $\frac{1}{2d}$ higher than the second term. Hence, we lower bound by considering one scenario where the first term is non-zero.

Continuing, we can further split the indicator function into cases when each term in the argument of the indicator function is positive.

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},y}\sigma\left(\frac{1}{2d}\sum_{j=1}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + b_i\right)x_1 \\
&\quad\geq \frac{1}{2d}\mathbb{E}_{\mathbf{x},y}\mathbb{I}\left(\frac{1}{2d}\sum_{j=2}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + b_i \geq 0\right) \\
&\quad\geq \frac{1}{2d}\mathbb{E}_{\mathbf{x},y}\mathbb{I}\left(\sum_{j=2}^{d} x_j \geq 0\right)\mathbb{I}\left(\sum_{j=d+1}^{n} x_j \geq 0\right)\mathbb{I}\left(b_i \geq 0\right) \\
&\quad\geq \frac{1}{8d}\mathbb{I}\left(b_i \geq 0\right).
\end{aligned}$$

From Equation (4), we then have

$$\mathbb{E}_{\mathbf{x},y}(f_{\mathcal{T}}^{(1)})_i(\mathbf{x},y)x_1 \geq \frac{1}{4d}\mathbb{I}\left(b_i \geq 0\right) + \mathcal{O}(\tau_g n \left|\zeta_{d-1}\right|^{-1}).$$

As b_i has been kept at random initialization and thus is a random variable selected from the set $\{-1+\frac{1}{d}, \cdots, 1-\frac{1}{d}\}$, with probability $\frac{1}{2}$, $\mathbb{I}\left(b_i \geq 0\right)$. This implies, w.p. atleast $1/2$ w.r.t. a neuron's bias initialization, $\mathbb{E}_{\mathbf{x},y}(f_{\mathcal{T}}^{(1)})_i(\mathbf{x},y)x_1 \geq \frac{1}{4d} + \mathcal{O}(\tau_g n \left|\zeta_{d-1}\right|^{-1})$. The final bound comes from the fact that $f_{\mathcal{T}}(\mathbf{x},y) = \frac{1}{m}\sum_{i=1}^{m}(f_{\mathcal{T}}^{(1)})_i(\mathbf{x},y) \geq \frac{1}{8d} + \mathcal{O}(\tau_g n \left|\zeta_{d-1}\right|^{-1}) + \mathcal{O}(m^{-1/2})$, where we apply a hoeffding's inequality to bound the error term.

Out of support correlations: Similar to the Equation (4), we have for $u \notin S$,

$$\begin{aligned}
\mathbb{E}_{\mathbf{x},y}(f_{\mathcal{T}}^{(1)})_i(\mathbf{x},y)x_u = \mathbb{E}_{\mathbf{x},y}\sigma&\left(\frac{1}{2d}\sum_{j=1}^{d} x_j + \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + \langle \boldsymbol{v}_i, \mathbf{x}\rangle + b_i\right)x_u \\
&- \sigma\left(-\frac{1}{2d}\sum_{j=1}^{d} x_j - \frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j + \langle \boldsymbol{v}_i, \mathbf{x}\rangle + b_i\right)x_u. \qquad (5)
\end{aligned}$$

However, we observe that the influence of x_u in each of the terms is bounded by $\frac{1}{d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}$. Consider the first term; the argument for the second term is similar. We can again ignore $\langle \boldsymbol{v}_i, \mathbf{x}\rangle$ incurring an error of $\mathcal{O}(\tau_g n\,|\zeta_{d-1}|^{-1})$.

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},y}\sigma\left(\frac{1}{2d}\sum_{j=1}^{d}x_j+\frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1}^{n}\operatorname{sign}(w_{ij}^{(0)})x_j+b_i\right)x_u\\
&\quad=\mathbb{E}_{\mathbf{x},y:x_u=+1}\sigma\left(\frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\operatorname{sign}(w_{iu}^{(0)})+\frac{1}{2d}\sum_{j=1}^{d}x_j+\frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1\to n;j\neq u}\operatorname{sign}(w_{ij}^{(0)})x_j+b_i\right)\\
&\quad-\mathbb{E}_{\mathbf{x},y:x_u=-1}\sigma\left(-\frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\operatorname{sign}(w_{iu}^{(0)})+\frac{1}{2d}\sum_{j=1}^{d}x_j+\frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1\to n;j\neq u}\operatorname{sign}(w_{ij}^{(0)})x_j+b_i\right)\\
&\quad=\mathbb{E}_{\mathbf{x},y}\frac{C(\mathbf{x})}{d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\operatorname{sign}(w_{iu}^{(0)})\mathbb{I}\left(\frac{1}{2d}\sum_{j=1}^{d}x_j+\frac{1}{2d}\frac{\zeta_{d+1}}{|\zeta_{d-1}|}\sum_{j=d+1\to n;j\neq u}\operatorname{sign}(w_{ij}^{(0)})x_j+b_i\geq 0\right),
\end{aligned}$$

where $C(\mathbf{x})\in\{1,2\}$ denotes a function that depends on $\mathbf{x}$. The final step follows from a first order taylor expansion of σ. The magnitude can hence be bounded by $\frac{1}{d}\frac{|\zeta_{d+1}|}{|\zeta_{d-1}|}$. This can be bounded by $\frac{1}{dn}$ (section 5.3, (O'Donnell, 2014)). □

B.3. Second stage analysis for the teacher

Lemma B.6 (Second stage Training, cf. Theorem 4 in Barak et al. (2022))**.** *Fix $\epsilon,\delta>0$. Suppose $m\geq\Omega(2^d d\log(d/\delta))$, $n\geq\Omega\left(d^4\log(dn/\epsilon)\right)$. Furthermore, suppose $B_1\geq\Omega(|\zeta_{d-1}|^2 d^2\log(dn/\epsilon))$ s.t. the weights satisfy the conditions in Theorem B.2 with $\tau_g=\mathcal{O}(|\zeta_{d-1}|\,d^{-1}n^{-1/2})$ after the first phase. Then after $T_2=\Omega(mn^2d^3/\epsilon^2)$ steps of training with batch size $B_2=1$, learning rate $\eta_2=4d^{1.5}/(n\sqrt{m(T_2-1)})$ and decay $\lambda_2=0$, we have with expectation over the randomness of the initialization and the sampling of the batches:*

$$\min_{t\in[T_2]}\mathbb{E}\left[L_{\theta^{(t)}}(\mathbf{x},y)\right]\leq\epsilon.$$

Thus, the minimal sample complexity to reach a loss of ϵ is given by

$$\begin{aligned}
T_1\times B_1+T_2\times B_2&=\Theta(|\zeta_{d-1}|^2 d^2\log(dn/\epsilon))+\Theta(mn^2d^3/\epsilon^2)\\
&=\Theta(n^{d-1}d^2\log(nd/\epsilon)+2^d n^2 d^4\epsilon^{-2}\log(d/\delta)).
\end{aligned}$$

Corollary B.7. *Under the conditions outlined in Theorem B.6, after T_2 steps of training in the second phase, if $t^\dagger$ denote the time step at which the model achieves the minimum loss, i.e. $t^\dagger:=\arg\min_{t\in[T_2]}\mathbb{E}\left[L_{\theta^{(t)}}(\mathbf{x},y)\right]$, then*

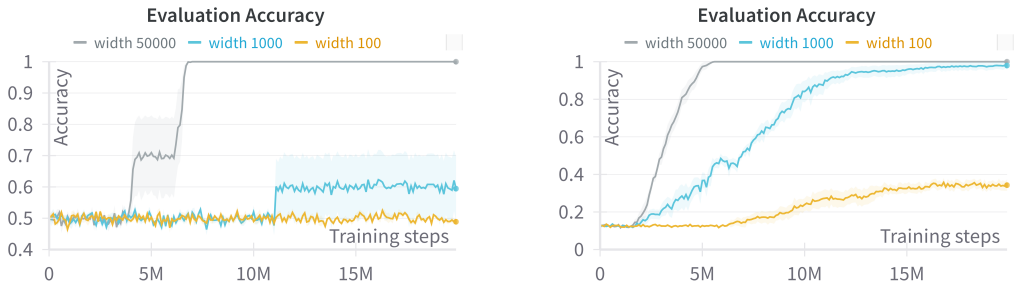
$$\mathbb{E}\left[f_{\mathcal{T}}^{(t^\dagger)}(\mathbf{x},y)x_i\right]\leq\epsilon,\ \textit{for all } i\in[n].$$

The proof follows from the fact that if the correlation along $y=\prod_{i\in S}x_i$ is large ($\geq 1-\epsilon$ as hinge loss is below ϵ), the correlations along other fourier basis functions will be small. Hence, depending on how saturated the model is, the signal along the support elements are small.

B.4. Sample complexity benefits with progressive distillation for the student:

Combining the results in Theorem B.7 and Theorem B.2, we have the final result.

Theorem B.8 (Sample complexity benefits with progressive distillation)**.** *Suppose we have a teacher model that has been trained with 2-stage training Algorithm 1 to loss $\mathcal{O}(n^{-c})$ for some constant $c\geq 1$, with its hyperparameters satisfying the conditions in Theorem B.2 and Theorem B.6. Suppose we train a student model $f_{\mathcal{S}}$ of size $\tilde{m}\geq\tilde{\Omega}(2^d d)$ but with two different strategies.*



(a) 6-sparse parity.

(b) 3-level, 5-variable features.

Figure 3. Models with a smaller width require more steps to learn for cross-entropy training (i.e. Equation (1) with $\alpha = 1$). The x-axis shows the training steps, and the y-axis shows the change in accuracy. Each line is the mean of 5 runs, with the shadow showing the standard error.

1. ***Progressive distillation:*** *Train for the first T_1 steps w.r.t. the teacher's logits at T_1 checkpoint. Then, train with the final teacher checkpoint in the second stage.*

2. ***Distillation:*** *Train with the final teacher checkpoint (by Theorem B.6) throughout training with any $\alpha \in (0, 1)$.*

Then,

1. *Under progressive distillation, the total sample complexity to reach a loss of ϵ with probability $1 - \delta$ is*

$$\Theta(d^2 \log(n\tilde{m}/\epsilon) + 2^d n^2 d^4 \epsilon^{-2} \log(d/\delta)).$$

2. *The necessary sample complexity under distillation is at least $\Omega(n^{\min(2c,d)})$.*

Proof. **Sample complexity for Progressive distillation:** Under progressive distillation, the label is given by $f_{\mathcal{T}}^{(T_1)}$ for the first T_1 steps. By Theorem B.5, $\mathbb{E}_{\mathbf{x},y} f_{\mathcal{T}}^{(T_1)}(\mathbf{x}, y) x_i \geq \Omega(d^{-1})$ for all $i \in S$, and $\mathbb{E}_{\mathbf{x},y} f_{\mathcal{T}}^{(T_1)}(\mathbf{x}, y) x_i \leq \mathcal{O}((dn)^{-1})$ for all $i \notin S$. With symmetric initialization of a_i's, we can show that $\mathbb{E}_{\mathbf{x},y} f_{\mathcal{T}}^{(T_1)}(\mathbf{x}, y) = 0$. Thus,

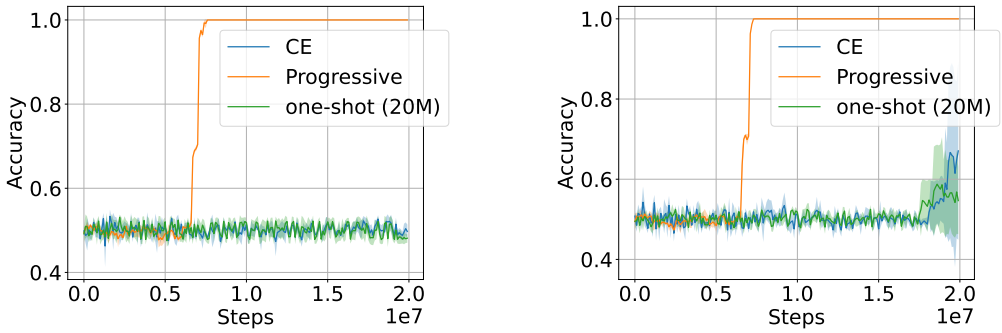
$$f_{\mathcal{T}}^{(T_1)}(\mathbf{x}, y) = \sum_{j=1}^{d} c_j x_j + \sum_{j=d+1}^{n} c_j x_j + \text{ higher-order polynomials },$$

with $|c_j| \geq \Omega(1/d)$ for $j \in S$ and $|c_j| \leq \mathcal{O}((dn)^{-1})$, for $j \notin S$. Since $\sum_{j=1}^{d} c_j x_j$ are of complexity 1, we can modify Theorem B.2 (specifically, Theorem B.4) to show that with appropriate learning rate, we only require a batch size of $B_1 \geq \Omega(n^2 \log(nm/\delta))$ to get the fourier gap between the coordinates in support and out of support. Thus, the change in the necessary sample complexity for Theorem B.6 comes from the reduced sample complexity in the first phase.

Sample complexity for Distillation: On the other hand, for the teacher checkpoint with loss $\mathcal{O}(n^{-c})$, the correlation to the monomial terms in the support is bounded by $\mathcal{O}(n^{(-c)})$ (by Theorem B.7). If we want to learn from the correlations to the support, we need the number of samples to be at least $\Omega(n^{2c})$ as the gradient noise needs to be lower than $\mathcal{O}(n^{-c})$ (by Theorem B.4). To learn the support from the true label, we need the number of samples to be at least $\Omega(n^d)$ (by Theorem B.1). Hence, for the model to learn the support from a combination of the two components, it needs a sample complexity at least $\Omega(n^{\min(2c,d)})$. □

C. Learning hierarchical data

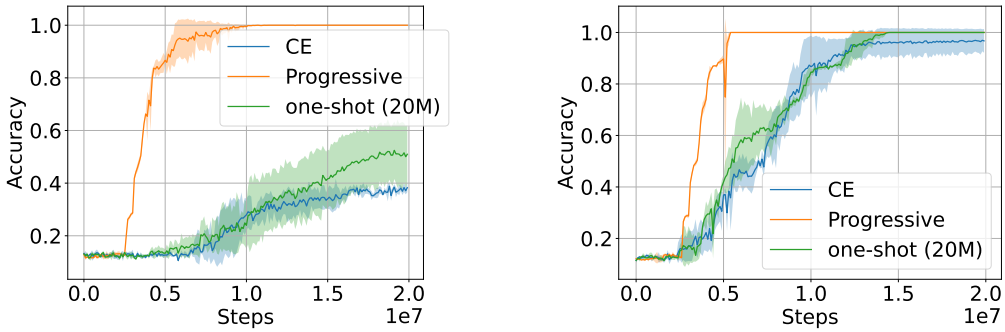
Formal definition: The input $\mathbf{x}$ is a boolean vector picked uniformly at random from the n-dimensional hypercube $\{\pm 1\}^n$, and the label $y \in [K]$ where $K := 2^D$ for some fixed $D \in \mathbb{N}$. The underlying labeling function for y follows a binary



(a) Width-100 student.

(b) Width-1000 student.

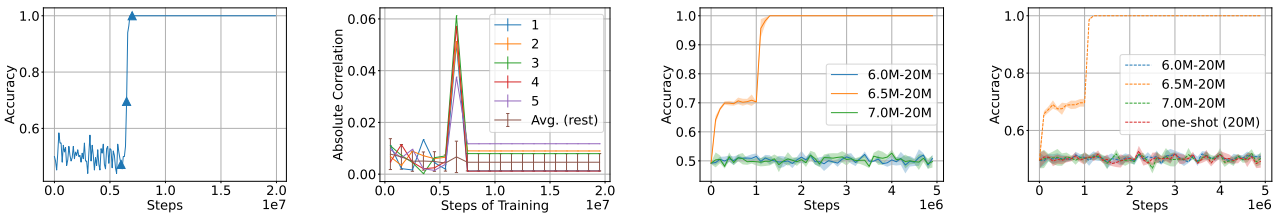
Figure 4. Experiments on 6-sparse parity. Progressive distillation helps student learn faster (Equation (1) with $\alpha = 0$), compared to one-shot distillation from a later checkpoint. The x-axis shows the training steps, and the y-axis shows the change in accuracy for learning 6-sparse parity. Each line is the mean of 3 runs, with the shadow showing the standard error. The green curve is for progressive distillation at 500k-step intervals; the yellow and red curves are for one-shot distillation from checkpoints at 10M and 20M steps, respectively. The teacher's temperature is 1.



(a) Width-100 student.

(b) Width-1000 student.

Figure 5. 8-way classification using a hierarchical decision tree of depth 3, with each node represented by 5-sparse parity. Progressive distillation helps student learn faster (Equation (1) with $\alpha = 0$), compared to one-shot distillation from a later checkpoint. The x-axis shows the training steps, and the y-axis shows the change in accuracy for learning 6-sparse parity. Each line is the mean of 3 runs, with the shadow showing the standard error. The green curve is for progressive distillation at 500k-step intervals; the yellow and red curves are for one-shot distillation from checkpoints at 10M and 20M steps, respectively. The teacher's temperature is 1.



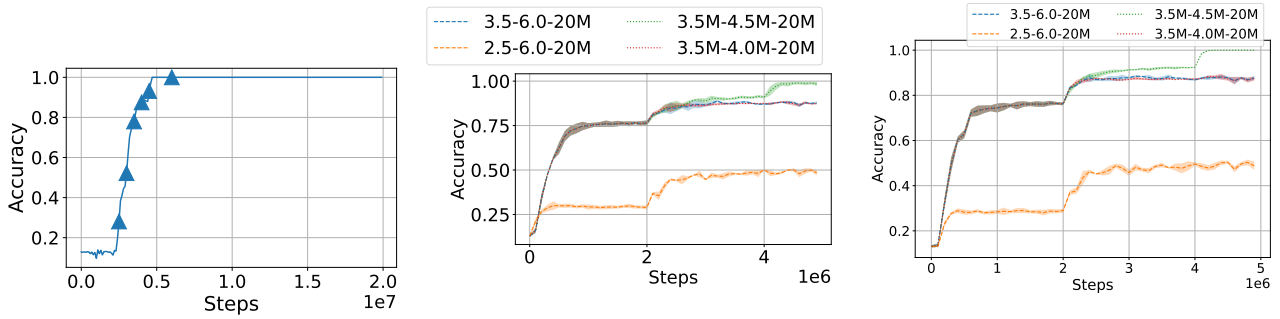
(a) Teacher accuracy behavior

(b) Correlations $\mathbb{E}_{\mathbf{x},y} f_{\mathcal{T}}(\mathbf{x}, y) x_i$ for $i \in S$ during training

(c) Width 100: 2-shot Distillation ($\tau = 1$)

(d) Width 100: 2-shot Distillation ($\tau = 10^{-4}$)

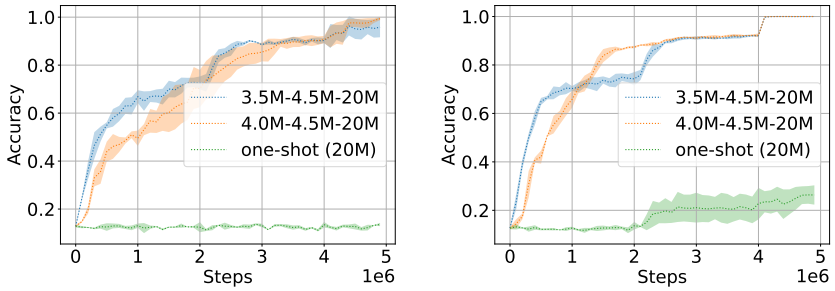
Figure 6. Continued from Figure 2. (a, b) have been repeated for the ease of presentation. (c, d) show $2-$shot distillation results for a student of width 100. (c) Teacher's checkpoint during phase transition ($6.5M$) helps 2-shot Distillation performance to converge to 100% accuracy, while other checkpoints don't. (d) Even with extremely low temperature, the benefit of the phase transition checkpoint persists, suggesting that the monomial curriculum, not regularization, is the key to the success of progressive distillation. For the 2-shot Distillation in (c, d), the student is trained with an intermediate checkpoint for $1M$ steps, followed by distillation from the final teacher checkpoint until the end of training.



(a) Teacher accuracy behavior

(b) Width 100: 3-shot distillation ($\tau = 1.0$)

(c) Width 1000: 3-shot distillation ($\tau = 1.0$)



(d) Width 100: 3-shot distillation ($\tau = 10^{-4}$)

(e) Width 1000: 3-shot distillation ($\tau = 10^{-4}$)

Figure 7. Setting: Depth-3 tree with 5-variable features. 3-shot Distillation from 3 checkpoints; 2 intermediate teacher checkpoints are used each for $2M$ steps, and then the final checkpoint is used till end of training. **Observations:** (a) Teacher shows a phase transition in accuracy during training. 6 candidate checkpoints for $3-$shot Distillation have been marked by triangles, out of which 2 are selected in each setting. The checkpoint at $6M$ lies outside the phase transition of the teacher. (b, c): We show the behavior of a few representative settings. Two main observations: (1) Selecting only a single checkpoint during the phase transition of the teacher is sub-optimal, (2) 2 checkpoints during the stage transition suffice to train the student to 100% accuracy, however the performance can heavily depend on their selection. Figure 8 shows that the teacher learns the low-level features at $4.5M$ checkpoint, making it crucial for distillation. (d, e): Even with extremely low temperature, the benefit of the phase transition checkpoint persists, suggesting that the monomial curriculum, not regularization, is the key to the success of progressive distillation.

decision tree of depth D, whose leaves correspond to class labels. The branching at a node depends on a sparse parity problem. An example visualization is provided in Figure 9.

More formally, the nodes in the decision tree are represented by a set of sparse parity problems $\mathbb{S} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_{K-1}\}$, where $\mathcal{T}_j$ is determined by product of a subset of size d variables selected from the dimensions of the input $\mathbf{x}$ (e.g. $x_1 x_2 \cdots x_5$ for $d = 5$). An input $\mathbf{x}$ belongs to the class $i \in [K]$ iff

$$\left[\prod_{j=1}^{D} \mathbb{I}[\mathcal{T}_{v_j^{(i)}}(\mathbf{x}) > 0]\right] > 0.$$

Here, $v_1^{(i)}, \cdots v_D^{(i)}$ denote the features in $\mathbb{S}$ that lie on the path joining the root of the decision tree to the leaf representing the label i.

Experiment Setup: In this section, we focus on 8-way classification, where the data is generated by a tree of depth 3. Each feature in $\mathbb{S}$ is given by a product of 5 variables. We keep the variables distinct in each feature, i.e., $\mathcal{T}_1 = x_1 x_2 \cdots x_5$, $\mathcal{T}_2 = x_6 x_7 \cdots x_{10}$ and so on.

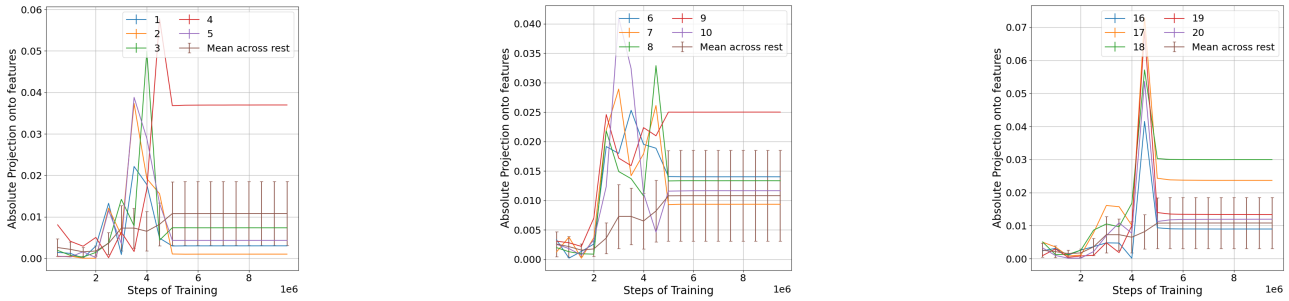


Figure 8. Projection onto the features for *intermediate* teacher checkpoints. The projections for a depth-3 tree with 5-variable features. The 3 plots show one feature in each of the 3 levels of the tree.

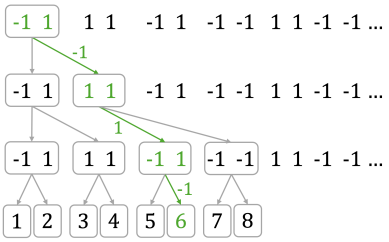


Figure 9. An illustration of hierarchical data generation, for a 3-level tree with 5 variables per feature. A feature corresponds to a tree node, each marked by a rectangle. The product of the binary variables in a feature determines which child to take: the left child is chosen if the product evaluates to 1, and the right child is chosen if the product is -1. The path leading to the label and the values evaluated in the corresponding nodes are highlighted in green.