# Strong Teacher Not Needed? On Distillation in LLM Pretraining

**Taiming Lu**
Princeton University
taiminglu@princeton.edu

**Zhuang Liu**
Princeton University
zhuangl@princeton.edu

## Abstract

Knowledge distillation generally assumes a strong-to-weak relationship where stronger teachers yield better students. In this work, we examine this assumption about distillation in large language model pretraining. By varying architecture sizes and training token budgets, we create strong-to-weak, same-level, and weak-to-strong teacher-student relationships. We study distillation's effectiveness under these relationships with different mixes between language and distillation loss. Three findings emerge: (1) with proper loss mixing, weak-to-strong and same-level distillation improves over standard pretraining, where even small and undertrained teachers benefit large students; (2) making the teacher stronger can lead to saturated or even reversed gains; (3) distillation improves generalization (out-of-distribution, downstream) more readily than in-domain fitting. Our results challenge the common belief that distillation pretraining always requires a strong teacher.

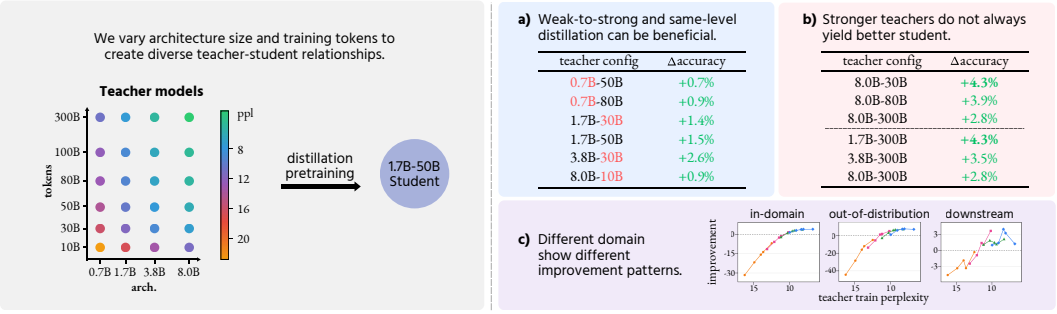


Figure 1: **A strong teacher is not always needed in distillation pretraining.** We vary architecture size and training tokens to create diverse teacher-student relationships. We have three key findings. **a)** Weak-to-strong and same-level distillation can be beneficial: teachers with smaller architecture or fewer tokens (marked red in figure) than the student still yield improvement. **b)** Stronger teachers do not always yield better students: at high token budgets, same-size teachers outperform larger ones; at large sizes, teachers trained with fewer tokens outperform those trained with more. **c)** Different evaluation domains show different patterns: in-domain improvement follows a universal trend and is hardest to improve, while out-of-distribution and downstream exhibit architecture-specific or noisy patterns and are easier to improve, suggesting distillation pretraining transfers useful knowledge for generalization instead of fitting.

## 1 Introduction

A long-standing assumption governs knowledge distillation: a stronger teacher yields a better student [22, 6]. When a large model is compressed into a smaller one, the more capable the teacher,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the more knowledge there is available to transfer [12, 9]. Across a decade of work spanning both vision [51, 66, 41] and language modeling [53, 24, 56, 59, 26], this strong-to-weak framing has shaped teacher selection [8], loss design [19, 44, 63], and the broader premise that distillation serves primarily as a compression technique [36, 69].

Our experiments suggest this account is incomplete. In large language model (LLM) pretraining, we find that weaker teachers can in fact improve stronger students, and that continuing to strengthen a teacher, through additional parameters or additional training tokens, can degrade rather than improve the resulting student, a reversal we observe consistently across architectures and training budgets.

The stakes extend well beyond theoretical interest. As LLMs continue to scale, two practical scenarios make the teacher-student relationship increasingly consequential. First, when training frontier models [1, 48, 16], the best available teacher is often drawn from the previous generation, trained with a smaller architecture or less compute than the target student [7]. Second, alongside the push toward larger models, demand persists for small, efficient systems [45, 31] capable of serving lightweight tasks at low latency [28].

Under the conventional strong-to-weak view, the former scenario forecloses distillation entirely, while the latter [32, 69] reduces teacher selection to a single rule: the larger and more trained, the better [36]. Whether these assumptions hold determines both how broadly distillation can be applied in LLM development and how the teacher should be chosen in practice.

Despite its centrality, the strong-to-weak view has remained largely unchallenged in the context of LLM pretraining [6]: weaker teachers are presumed unable to meaningfully guide stronger students, and scaling up the teacher is presumed to monotonically improve the student. Prior work on LLM distillation [26, 56, 59, 60, 53, 24] has not systematically examined these assumptions across the full range of teacher configurations that practitioners encounter in practice. We confront them directly by asking two central questions:

*Can weak-to-strong distillation improve the student?*

*Does a stronger teacher always yield a better student?*

To address these questions, we conduct a systematic study of teacher-student relationships (see Figure 1). We vary teacher and student architecture sizes, teacher training token budgets, and the loss mixing coefficient, spanning strong-to-weak, same-level, and weak-to-strong configurations. We measure distillation effectiveness on in-domain perplexity, out-of-distribution perplexity, and downstream accuracy. Three findings emerge, each departing from conventional expectations.

- **Weak teachers can improve strong students.** With appropriate loss mixing, teachers smaller and less trained than the student still yield measurable gains; even same-level distillation, in which teacher and student share both architecture and token budget, consistently improves performance.
- **Stronger teachers do not always produce better students.** Architecture mismatch and teacher overtraining can degrade transfer even when the teacher attains a lower training loss.
- **Generalization improves more readily than in-domain fit.** Out-of-distribution perplexity and downstream accuracy exhibit larger and more consistent gains than in-domain perplexity, and their improvement depends beyond teacher loss, notably architecture match and training token.

Collectively, these findings unsettle the prevailing account of distillation and offer concrete implications for which teacher to select and which loss-mixing to use under realistic compute constraints.

## 2 Background

**Autoregressive language modeling.** Given a token sequence $x = (x_1, \ldots, x_T)$ drawn from a pretraining corpus $\mathcal{D}$, a student model with parameters $\theta_S$ defines a next-token distribution $p_{\theta_S}(\cdot \mid x_{<t})$, and a teacher model with parameters $\theta_T$ defines $p_{\theta_T}(\cdot \mid x_{<t})$. Standard pretraining learns $\theta_S$ by maximizing the likelihood of the observed next token:

$$\mathcal{L}_{\mathrm{lm}}(\theta_S) = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{T}\sum_{t=1}^{T} -\log p_{\theta_S}(x_t \mid x_{<t})\right]. \tag{1}$$

**Knowledge distillation for language modeling.** In distillation pretraining [22], the target is no longer a one-hot vector but the teacher's relative preferences among all tokens. The teacher

provides a full distribution over the vocabulary for the same context $x_{<t}$. Let $q_t(\cdot) := p_{\theta_T}(\cdot \mid x_{<t})$ and $p_t(\cdot) := p_{\theta_S}(\cdot \mid x_{<t})$. Student distribution $p_t$ will try to match the teacher distribution $q_t$ by minimizing the forward KL divergence:

$$\mathcal{L}_{\text{kd}}(\theta_S; \theta_T) = \mathbb{E}_{x \sim \mathcal{D}} \left[ \frac{1}{T} \sum_{t=1}^{T} \text{KL}(q_t \,\|\, p_t) \right]. \tag{2}$$

**Mixing ground-truth and teacher supervision.** Distillation typically combines the standard language modeling loss with the distillation loss [22], where $\alpha \in [0, 1]$. The mixing coefficient $\alpha$ controls the trade-off: $\alpha = 0$ recovers standard pretraining, while larger $\alpha$ places more weight on matching the teacher distribution. In practice, $\alpha$ is a key hyperparameter because it determines how much the student follows the teacher compared to the ground-truth data signal. We refer to this as the mixed loss and use it for all distillation pretraining:

$$\mathcal{L}(\theta_S; \theta_T) = (1 - \alpha)\, \mathcal{L}_{\text{lm}}(\theta_S) + \alpha\, \mathcal{L}_{\text{kd}}(\theta_S; \theta_T), \tag{3}$$

# 3 Experimental Design

## 3.1 Teacher-Student Configurations

**Architecture sizes.** We use the Llama3 [1] architectures. To enable systematic comparison across sizes, we create four architecture variants (see §B), resulting in four model sizes: 0.7B, 1.7B, 3.8B, and 8.0B. Given a student of a particular size, the teacher can be larger, the same size, or smaller.

**Training token budgets.** Teachers are pretrained on varying amounts of data: 10B, 30B, 50B, 80B, 100B, or 300B tokens. Students are trained on all 50B tokens for distillation pretraining. This creates another dimension of variation: the teacher sees more, the same, or fewer tokens than the student.

**Teacher-student relationships.** Together, architecture size and training tokens create a spectrum of teacher-student relationships. We categorize them along three dimensions:

- Architecture: large-to-small, same-size, or small-to-large,
- Tokens: more-to-less, same-number, or less-to-more,
- Train loss: lower-to-higher or higher-to-lower, defined by the teacher's final train loss compared to the student's standard pretraining baseline loss.

These dimensions do not always align. A smaller architecture trained on more tokens may achieve lower train loss than a larger one trained on fewer tokens. A same-size teacher trained longer may outperform a larger one trained briefly. This allows us to disentangle the effects of architecture capacity from training compute in knowledge transfer. More broadly, we refer to a teacher-student relationship as *strong-to-weak* when a teacher is stronger than the student (by any dimensions above), *same-level* if identical, and *weak-to-strong* for the opposite. A single relationship may be strong-to-weak in one dimension and weak-to-strong in another.

## 3.2 Training Details

**Data and tokenization.** All models are trained on FineWeb-Edu [42], a high-quality web corpus. Teacher pretraining and student distillation pretraining draw from the same data pool with different random seeds, so the sequences seen during training may overlap but are not identical. For each run, no data is repeated during training so each sequence is seen at most once per run.

**Hyperparameters.** We use an 8192-token context length and a 256-sequence batch size. We train with AdamW using $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 0.1. The peak learning rate is $3 \times 10^{-4}$ with a cosine decay to 10% of the peak, linear warm-up for the first 5% of steps, and bf16 precision.

**Distillation pretraining.** The teacher model is kept frozen and provides logits. We use a temperature of $\tau = 1$ for the knowledge distillation loss to apply no scaling to the logits. For each teacher-student configuration, we search over the loss mixing coefficient $\alpha \in \{0.2, 0.4, 0.5, 0.6, 0.8, 1.0\}$.

**Baseline.** The standard pretraining baseline for each student corresponds to training with $\alpha = 0$ (language modeling loss only) under the same data ordering and random seed as distillation. This ensures that any improvement is attributable to the teacher signal rather than confounding factors.

| | in-domain ppl (↓) | | | | out-of-distribution ppl avg. (↓) | | | | downstream acc avg. (↑) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B |
| → *standard pretrained teachers* | | | | | | | | | | | | |
| 10B | 21.2 | 16.8 | 14.4 | 12.8 | 39.8 | 26.3 | 22.0 | 18.5 | 34.9% | 37.0% | 38.5% | 39.4% |
| 30B | 16.3 | 13.1 | 11.2 | 10.0 | 23.6 | 17.5 | 14.4 | 12.8 | 37.8% | 39.4% | 41.5% | 42.7% |
| 50B | 14.9 | 12.1 | 10.3 | 9.1 | 20.6 | 15.6 | 13.0 | 11.7 | 38.1% | 39.9% | 42.9% | 43.7% |
| 80B | 14.0 | 11.5 | 9.6 | 8.4 | 18.4 | 14.5 | 11.9 | 11.0 | 38.5% | 41.9% | 43.6% | 44.9% |
| 100B | 13.7 | 11.2 | 9.4 | 8.1 | 17.8 | 13.9 | 11.6 | 10.6 | 38.8% | 41.9% | 43.7% | 45.2% |
| 300B | 12.5 | 10.2 | 8.3 | 6.8 | 16.1 | 12.8 | 10.9 | 10.1 | 40.1% | 43.2% | 45.4% | 46.6% |
| → *standard pretrained student baseline* | | | | | | | | | | | | |
| 50B | 14.9 | 12.2 | 10.3 | 9.0 | 20.6 | 15.7 | 13.0 | 11.7 | 38.1% | 40.1% | 42.2% | 43.5% |

Table 1: **Performance of standard pretrained teacher models and student baselines.** Rows indicate training token budget; columns indicate architecture size. In our main experiments, student distillation pretraining (and thus the baselines) is fixed to 50B tokens. We report in-domain perplexity (ppl) on held-out FineWeb-Edu (lower is better), average out-of-distribution perplexity across 11 corpora (lower is better), and average downstream accuracy (acc) across 15 benchmarks (higher is better). Teacher performance consistently improves with both architecture size and training tokens, with minor variance in downstream accuracy. View full results at §I.

### 3.3 Evaluations

**In-domain perplexity.** We measure in-domain perplexity (id ppl) on a held-out test split from training dataset FineWeb-Edu, which was not seen during training. Lower perplexity indicates better fit to the training distribution, for how well the student models same distribution as training data.

**Out-of-domain perplexity.** We also measure out-of-distribution perplexity (ood ppl) on 11 high-quality corpora: Wikitext-103 [33], C4 [49], GSM8K [14], DM Mathematics [55], HumanEval [11], CodeSearchNet [23], arXiv [15], CNN-DailyMail [21], ECHR [10], PubMedQA [25], XQuAD [3]. These corpora span diverse domains, including web, news, scientific, math, code, biomedical, legal, and multilingual documents and QA, providing broad coverage of distribution shifts. This evaluates how well the model generalizes beyond the training corpus.

**Downstream accuracy.** We evaluate on 15 downstream accuracy-based benchmarks (downstream acc): MMLU [20], ARC-Easy [13], SciQ [61], OpenBookQA [34], MathQA [2], TruthfulQA [29], ANLI-R1 [39], CommonsenseQA [57], HellaSwag [67], PIQA [5], WinoGrande [52], Social IQa [54], LogiQA 2.0 [30], MedMCQA [40], and RACE [27]. These benchmarks cover broad knowledge, science, math, and medical QA, commonsense and logical reasoning, and reading comprehension to offer a diverse view of downstream transfer, for tasks beyond next-token prediction.

**Measuring improvement.** For each evaluation type, we compute improvement as the percentage change relative to the standard pretraining baseline. Due to the high number of benchmarks within each evaluation type, we report the average for brevity, where we average the percentage improvements across benchmarks rather than averaging raw scores, as they have different scales. Across multi-seed baseline runs (§H), in-domain perplexity standard deviation is below 0.5% of the mean and downstream accuracy standard deviation is below 0.01, so reported improvements (typically 3–9%) exceed seed noise by an order of magnitude. Full implementation details are in §C.

## 4 Distillation Under Different Teacher-Student Relationships

We train teachers across four architecture sizes (0.7B, 1.7B, 3.8B, 8.0B) and six token budgets (10B, 30B, 50B, 80B, 100B, 300B), yielding 24 teachers spanning undertrained small to well-trained large regimes. Table 1 confirms performance improves with both architecture size and training tokens, and reports the 50B-token student baselines used as references.

We fix the student to 1.7B at 50B tokens and pair it with all 24 teachers, selecting the optimal loss mixing coefficient $\alpha$ per teacher-student configuration to study distillation's maximum strength. We report the best $\alpha$ per evaluation type; a single best $\alpha$ across all types yields the same findings (§E). Average improvement is reported, with per-benchmark results in §I and training curves in §F. Pure distillation ($\alpha = 1.0$) is generally worse than mixed loss (§D).

| | %Δ in-domain ppl (↑) | | | | avg. %Δ out-of-distribution ppl (↑) | | | | avg. %Δ downstream acc (↑) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B |
| 10B | -2.9% (0.2) | -1.4% (0.2) | -0.4% (0.2) | **0.3%** (0.2) | -3.2% (0.2) | -1.1% (0.2) | 1.2% (0.2) | **1.6%** (0.4) | -0.5% (0.2) | -0.4% (0.2) | -0.8% (0.5) | **0.9%** (0.4) |
| 30B | -1.2% (0.2) | 0.5% (0.2) | 1.9% (0.5) | **3.2%** (0.6) | -0.5% (0.2) | 1.5% (0.2) | 4.4% (0.5) | **5.4%** (0.5) | 1.3% (0.2) | 1.4% (0.2) | 2.6% (0.5) | **4.3%** (0.6) |
| 50B | -0.3% (0.2) | 1.0% (0.4) | 3.0% (0.6) | **3.9%** (0.8) | 1.6% (0.2) | 3.5% (0.4) | 6.1% (0.5) | **7.3%** (0.8) | 0.7% (0.2) | 1.5% (0.2) | **3.4%** (0.6) | 3.2% (0.8) |
| 80B | 0.1% (0.2) | 2.0% (0.4) | 3.6% (0.6) | **4.3%** (0.8) | 2.6% (0.4) | 5.3% (0.6) | **7.7%** (0.8) | 7.6% (0.6) | 0.9% (0.4) | 1.8% (0.2) | 3.6% (0.8) | **3.9%** (1.0) |
| 100B | 0.4% (0.2) | 2.3% (0.5) | 3.9% (0.8) | **4.2%** (0.8) | 3.1% (0.2) | 6.0% (0.6) | 7.6% (0.8) | **8.9%** (0.8) | 0.7% (0.4) | 1.9% (0.5) | **3.3%** (0.6) | 3.2% (1.0) |
| 300B | 0.9% (0.2) | 3.3% (0.6) | **4.3%** (0.8) | 4.0% (0.8) | 4.4% (0.6) | **7.6%** (0.6) | 7.4% (0.8) | 7.2% (0.6) | 1.5% (0.6) | **4.3%** (0.8) | 3.5% (0.5) | 2.8% (0.8) |

Table 2: **Distillation improvement over standard pretraining baseline under the best loss mixing coefficient $\alpha$.** The student is 1.7B trained on 50B tokens. Rows indicate teacher token budget; columns indicate teacher architecture size. Each cell shows percentage improvement (positive is better) with the corresponding best $\alpha$ in parentheses. Bold indicates best within row (same tokens, different architectures); underline indicates best within column (same architecture, different tokens). Even weak teachers (smaller architecture or fewer tokens than the student) can improve distillation with proper $\alpha$ selection. However, the strongest teachers do not always yield the best results: at 300B tokens, smaller teacher architectures can outperform larger ones. Per-benchmark results are in §I.

Table 2 and Figure 2 show that most teacher configurations yield positive improvement across all three evaluation types, with only the weakest teachers (0.7B and 1.7B at 10B tokens) degrading performance. Improvement generally increases with teacher size, training tokens, and teacher's in-domain perplexity (a capability proxy). Notable deviations from general expectations emerge.

### 4.1 Weak-to-Strong and Same-Level Distillation

**Finding 1:** With proper language modeling and distillation loss mixing, even weak-to-strong and same-level distillation improves over standard pretraining.

Even teachers weaker than the student can improve distillation. In Figure 2, configurations to the left of the vertical dashed lines represent weak-to-strong distillation in that dimension. Across all three weak-to-strong scenarios (smaller architecture, fewer tokens, or higher train loss), most configurations yield positive improvement. Most strikingly, even teachers weaker in both architecture and tokens (thus weaker in in-domain ppl as well) can help: the 0.7B teacher at 30B tokens still achieves 1.3% downstream accuracy improvement. This challenges the assumption that distillation requires a strictly stronger teacher. The key is using a lower loss mixing coefficient, which balances the weaker teacher's signal with ground-truth supervision (§J.5).

An interesting special case is when teacher and student share the same architecture and budget (1.7B at 50B in our setup), differing only in seed. This config still improves the student across all metrics (+1.0% in-domain ppl, +3.5% out-of-distribution ppl, +1.5% downstream acc), suggesting that distillation transfers beyond what architecture and data alone capture. This aligns with Born-Again Networks [17] in vision, where same-level distillation helps despite no capacity advantage.

### 4.2 Strong-to-Weak Distillation

**Finding 2:** Stronger teachers do not guarantee better distillation and can degrade transfer.

Surprisingly, the highest downstream accuracy improvement comes from the same-size teacher with more training tokens, instead of a stronger teacher that is both larger and trained on more tokens. In strong-to-weak, contrary to expectation, we observe saturation and reversal at a stronger teacher.

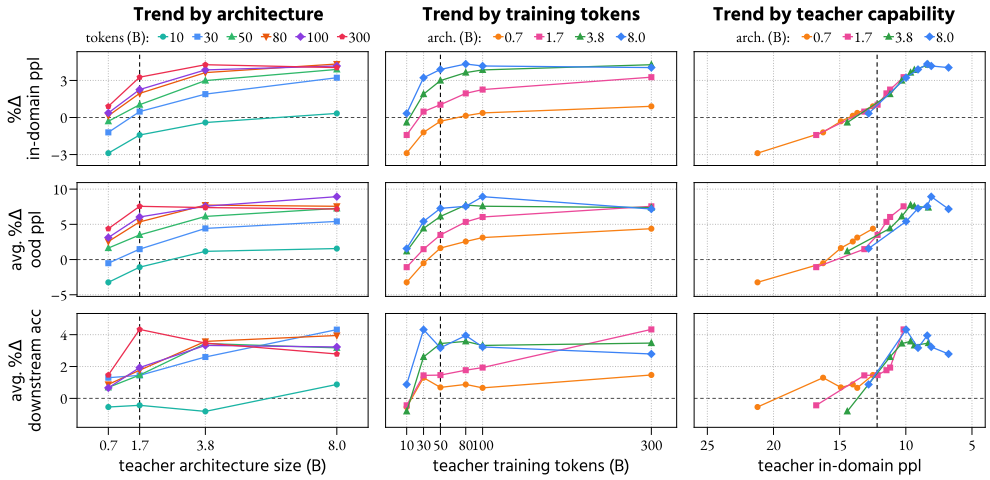


Figure 2: **Distillation improvement under best loss mixing coefficient $\alpha$.** Rows show evaluation types; columns show improvement along teacher architecture (left), training tokens (middle), and in-domain perplexity (right). Vertical dashed lines indicate the student's configuration (1.7B arch., 50B tokens, 12.2 ppl). Left of the dashed line represents weak-to-strong distillation in that dimension; right represents strong-to-weak. Improvement generally increases with teacher strength but saturates or reverses for strong teachers.

First, at high token budgets, larger architectures can underperform smaller ones. In Table 2, at 300B tokens, downstream improvement follows the trend of a smaller teacher distilling a better student: 1.7B (+4.3%) > 3.8B (+3.5%) > 8.0B (+2.8%), with the same-arch teacher outperforming larger ones. A similar pattern holds for out-of-distribution: 1.7B (+7.6%) > 3.8B (+7.4%) > 8.0B (+7.2%).

Second, at fixed architecture, training longer can hurt for large teachers. For the 8.0B teacher, downstream accuracy peaks at 30B tokens (+4.3%) and declines with further training: +3.9% at 80B, +3.2% at 100B, and +2.8% at 300B. The 3.8B teacher shows a similar pattern, peaking at 80B tokens (+3.6%) before declining. Out-of-domain perplexity follows the same trend: 8.0B teacher peaks at 100B tokens (+8.9%) and drops to +7.2% at 300B, and 3.8B teacher peaks at 80B. This suggests saturation at high teacher training budgets.

In contrast, when the teacher is same-architecture or smaller, more teacher training tokens monotonically lead to better distillation. These observations demonstrate that architecture mismatch between teacher and student can limit knowledge transfer, and that overtraining large teachers reduces the usefulness of their output distributions for smaller students. Thus, compatibility matters more than absolute teacher scale.

### 4.3 Performance by Domains

**Finding 3:** Distillation improves generalization (out-of-distribution, downstream) more easily than in-domain fitting.

We observe different trends across evaluation types. Out-of-domain perplexity and downstream accuracy are easier to improve than in-domain perplexity. In Table 2, several configurations that fail to improve in-domain perplexity still yield gains on the other two metrics. For example, the 0.7B teacher at 50B tokens degrades in-domain perplexity (-0.3%) but improves out-of-distribution perplexity (+1.6%) and downstream accuracy (+0.7%). The magnitude of improvement also differs: out-of-distribution perplexity shows the largest gains, reaching up to +8.9%, compared to +4.3% for both in-domain perplexity and downstream accuracy. Notably, in-domain and out-of-distribution evaluations are both perplexity tasks, yet distillation yields substantially better improvement on the curated out-of-distribution high-quality corpora. This pattern suggests that improvement on one evaluation type does not guarantee improvement on another. More broadly, distillation can transfer useful knowledge for generalization, even when it cannot improve fit to the training distribution.

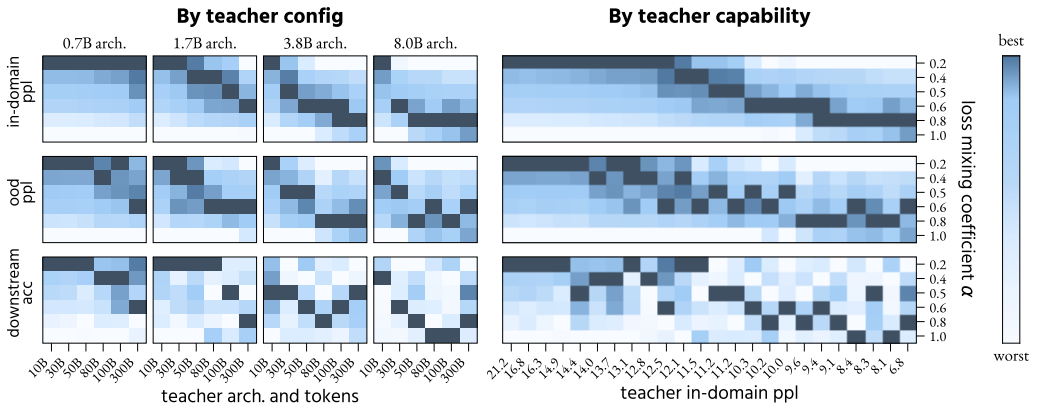


Figure 3: **Relative performance of $\alpha$ across teacher configurations.** Column-wise, each cell shows the normalized performance within each teacher's runs, where darker colors indicate better $\alpha$ for that teacher. Left panel groups teachers by architecture and training tokens; right panel orders all teachers by in-domain perplexity. All three evaluation types follow the same pattern, with downstream accuracy being noisier: for each teacher, performance increases gradually toward an optimal $\alpha$ then decreases, and stronger teachers favor higher $\alpha$.

# 5 The Role of Loss Mixing

## 5.1 Selecting the Optimal Mixing Coefficient

**Each teacher has an optimal $\alpha$.** Figure 3 (left) visualizes the relative performance of each $\alpha$ value across teacher configurations. For each teacher-evaluation pair, we apply min-max normalization across the six $\alpha$ values on evaluation improvements, where normalized $= (\text{value} - \min)/(\max - \min)$, so the best $\alpha$ maps to 100% (darkest) and the worst to 0% (lightest) for within-teacher comparisons. This normalization allows us to compare $\alpha$'s optimality patterns within each teacher regardless of absolute improvement magnitudes. Across all three evaluation types, the heatmap reveals a consistent pattern: moving from $\alpha = 0.2$ to $\alpha = 1.0$, performance rises toward an optimal value then falls. For perplexity evaluations, the trend is smooth and consistent, while downstream accuracy shows more variance. This smoothness suggests that optimal $\alpha$ reflects a true optimum rather than noise.

**Optimal $\alpha$ correlates with teacher strength.** Figure 3 (right) reorders teachers by their in-domain perplexity, revealing a clear trend: weaker teachers favor lower $\alpha$, while stronger teachers favor higher $\alpha$. Figure 4 directly confirms this pattern. For weaker teachers, low-$\alpha$ points cluster at the top (better improvement), whereas for stronger teachers, high-$\alpha$ points dominate. Intuitively, weaker teachers provide less reliable output distributions, so the student benefits from retaining more ground-truth supervision via lower $\alpha$ (§J.3).

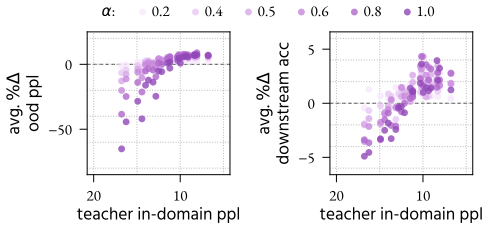


Figure 4: **Distillation improvement versus teacher capability by $\alpha$.** Each point represents one teacher-alpha combination. For weaker teachers (higher perplexity, left), lower $\alpha$ values yield better improvement. For stronger teachers (lower perplexity, right), higher $\alpha$ values perform better.

**Stronger teacher falls short: a loss mixing perspective.** Interestingly, for 3.8B and 8.0B teachers, we observe the optimal $\alpha$ rising then falling along the token dimension. In downstream accuracy, for example, the 8.0B teacher's optimal $\alpha$ across token budgets (10B, 30B, 50B, 80B, 100B, 300B) is 0.4, 0.6, 0.8, 1.0, 1.0, 0.8; for the 3.8B teacher, it is 0.5, 0.6, 0.6, 0.8, 0.6, 0.5. As these large teachers train longer, optimal $\alpha$ first increases, but eventually decreases at the high token budgets. This pattern does not appear for same-architecture (1.7B) or smaller (0.7B) teachers, where optimal $\alpha$ increases monotonically with training tokens. This provides additional evidence for the counterintuitive strong-to-weak observation in §4: when large teachers become overtrained and too strong, the student needs

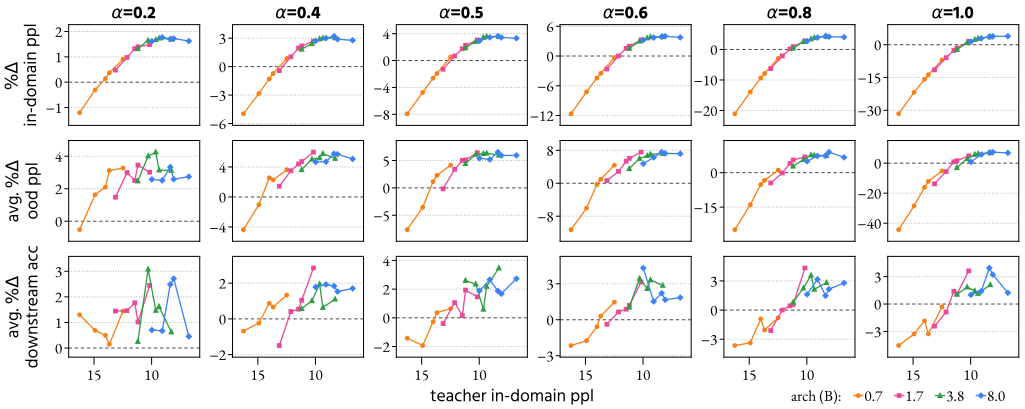


Figure 5: **Distillation improvement versus teacher in-domain perplexity under different loss mixing coefficient $\alpha$.** Rows show evaluation types; columns show the trend for increasing $\alpha$. Each line represents a teacher architecture, with points corresponding to different token budgets (10B-token teachers excluded for suboptimal and thus noisier performance). As $\alpha$ increases, trends become more obvious. For in-domain perplexity (top row), all architectures converge to a single trend: teacher loss alone predicts improvement regardless of architecture. For out-of-distribution perplexity (middle row), each architecture follows a separate trend, with smaller teachers outperforming larger ones even at higher loss levels. For downstream (bottom row), the upward general trend emerges with variance.

to reduce reliance on their signal. From a practical standpoint, the decreasing optimal $\alpha$ trend signals that investing in an even stronger teacher may yield diminishing returns or, in cases, even hurt.

### 5.2 Trends when Increasing Distillation Loss Weight

Previous sections tuned $\alpha$ per teacher configuration. We now examine how distillation behavior changes as $\alpha$ increases. For each $\alpha$ value, Figure 5 shows each evaluation type's improvement against teacher's in-domain perplexity to see whether any consistent trend emerges and how it evolves.

**Different evaluation types show different trends.** Figure 5 shows improvement versus teacher in-domain perplexity across $\alpha$. As $\alpha$ increases, trends become more pronounced across all evaluation types. For in-domain perplexity, all architectures converge to a single trend: improvement correlates with teacher loss regardless of architecture size. This pattern strengthens as $\alpha$ grows, with the four architecture lines nearly overlapping at high $\alpha$. This shows that given a teacher's loss, the student's in-domain improvement is directly predictable.

For out-of-distribution perplexity, each architecture follows a separate trend, with lines more horizontally separated than in the in-domain case, indicating that teacher loss and out-of-distribution improvement do not share the same relationship across architectures. For downstream accuracy, a general upward trend emerges but with considerable variance. These observations demonstrate that in-domain transfer closely follows teacher loss as we focus more on distillation, but out-of-distribution generalization and downstream performance depend on factors beyond teacher loss.

**Teacher train loss alone is a misleading metric for teacher quality.** The findings suggest that using teacher train loss as the sole indicator of teacher quality is misleading. While teacher train loss reliably indicates in-domain transfer, it fails to capture out-of-distribution and downstream performance. In Figure 5 (middle and bottom row), smaller teachers with higher train loss yield better improvement than larger teachers with lower train loss. This arises from the additional tokens the smaller teacher has seen during pretraining, which provides broader coverage of the data distribution despite a higher loss for a subset. Practitioners selecting teachers based on train loss therefore miss configurations that transfer better for generalization.

| student \ teacher | %Δ in-domain ppl (↑) | | | | avg. %Δ out-of-distribution ppl (↑) | | | | avg. %Δ downstream acc (↑) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B |
| 0.7B | 0.1% (0.2) | 1.8% (0.6) | 2.6% (0.8) | **2.8%** (0.8) | 0.5% (0.2) | 3.2% (0.6) | **5.4%** (0.6) | 5.3% (0.6) | 1.3% (0.6) | 2.1% (0.4) | **2.7%** (1.0) | 1.9% (1.0) |
| 1.7B | -0.3% (0.2) | 1.0% (0.4) | 3.0% (0.6) | **3.9%** (0.8) | 1.6% (0.2) | 3.5% (0.4) | 6.1% (0.5) | **7.3%** (0.8) | 0.7% (0.2) | 1.5% (0.2) | **3.4%** (0.6) | 3.2% (0.8) |
| 3.8B | -1.8% (0.2) | -0.3% (0.2) | 1.2% (0.4) | **2.8%** (0.5) | 0.3% (0.2) | 2.7% (0.2) | 4.7% (0.4) | **6.2%** (0.5) | 0.3% (0.2) | 0.1% (0.5) | 2.8% (0.4) | **3.0%** (0.5) |
| 8.0B | -3.2% (0.2) | -1.9% (0.2) | -0.6% (0.2) | **0.8%** (0.4) | 1.4% (0.2) | 3.4% (0.2) | 4.8% (0.4) | **7.1%** (0.5) | -0.3% (0.2) | 0.6% (0.2) | 1.8% (0.2) | **3.1%** (0.4) |

Table 3: **Distillation improvement with fixed 50B training tokens for both teacher and student.** Rows indicate student architecture; columns indicate teacher architecture. Diagonal cells represent same-level distillation; above-diagonal represent strong-to-weak; below-diagonal represent weak-to-strong. Strong-to-weak consistently improves all metrics. Weak-to-strong is effective out-of-distribution and downstream but struggles in-domain, especially with larger students.

## 6 Ablation: Generalization Across Student Sizes

We conduct additional experiments with varying student sizes. Our main experiments fix the student to 1.7B. We now vary both teacher and student architecture sizes while keeping training tokens fixed at 50B for both. This creates a symmetric setting where the only difference between teacher and student is the architecture size. Table 3 shows the results. The findings from our main experiments generalize across student sizes. Strong-to-weak distillation (above-diagonal) consistently improves all metrics across all student sizes, but the greatest improvement does not necessarily come with the maximum teacher-student size gap. Weak-to-strong distillation (below-diagonal) remains effective for out-of-distribution perplexity and downstream accuracy, but struggles for in-domain perplexity, particularly for larger students; the 8.0B student shows negative in-domain improvement with all smaller teachers. Same-level distillation (diagonal) yields positive improvement across all metrics and student sizes, confirming distillation transfers useful knowledge even without a teacher advantage. Experiments in §G ablate distillation tokens and architecture families to further support the conclusions.

## 7 Related Work

**Knowledge distillation for language models.** Knowledge distillation (KD) [22] transfers knowledge from a teacher to a student by training the student to match the teacher's output distribution. The teacher's soft predictions contain dark knowledge to guide the student, or relative similarities between classes absent in hard labels [38, 58]. Originally proposed for model compression in vision [6], KD has been widely adopted in LLMs, from masked LMs [56, 53, 24] to autoregressive LMs [59]. While recent industrial models primarily use teacher-generated synthetic data [1, 47, 48, 16], we revisit logit-level distillation in pretraining. Unlike prior work that focuses on compression with a fixed strong teacher, we systematically study how teacher size and compute affect distillation outcomes.

**Factors governing distillation success.** Distillation outcomes depend on design choices beyond teacher strength, including loss design [19, 44], logits processing [63], target generation [18, 43], and intermediate signal matching [51, 56]. In this work, we focus on the regime of teacher-student relations. In practice, industrial models increasingly rely on both weaker and stronger prior-generation or cross-family models as teachers [47, 48, 16], yet systematic studies on their relations in LLM pretraining remain limited. This work fills this gap and challenges common assumptions.

## 8 Conclusion

We systematically study knowledge distillation in LLM pretraining across a range of teacher-student relationships. Our findings challenge common assumptions by showing weak-to-strong distillation can help and stronger teachers do not always yield better students. Different evaluation types also show different patterns. Together, these results challenge the common belief.

**Acknowledgments**

This work is supported by the computational resources generously provided by Google's TPU Research Cloud program. This work was also performed using Princeton Research Computing resources, a consortium led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Research Computing at Princeton University. We also gratefully acknowledge the use of the Neuronic GPU computing cluster maintained by the Department of Computer Science at Princeton University. We thank Sachin Konan, Yida Yin, and Boya Zeng for helpful discussions.

## References

[1] AI@Meta. The llama 3 herd of models. *arXiv:2407.21783*, 2024.

[2] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *NAACL*, 2019.

[3] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *ACL*, 2020.

[4] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *CVPR*, 2022.

[5] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

[6] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006.

[7] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *ICML*, 2024.

[8] Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. Distillation scaling laws. In *ICML*, 2025.

[9] Sungmin Cha and Kyunghyun Cho. Why knowledge distillation works in generative models: A minimal working explanation. In *NeurIPS*, 2025.

[10] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in english. In *ACL*, 2019.

[11] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

[12] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *ICCV*, 2019.

[13] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457*, 2018.

[14] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

[15] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *NAACL*, 2018.

[16] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948*, 2025.

[17] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018.

[18] Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Miniplm: Knowledge distillation for pre-training language models. In *ICLR*, 2025.

[19] Changyi He, Yifu Ding, Jinyang Guo, Ruihao Gong, Haotong Qin, and Xianglong Liu. Da-kd: Difficulty-aware knowledge distillation for efficient large language models. In *ICML*, 2025.

[20] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021.

[21] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NeurIPS*, 2015.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[23] Hamel Husain, Hongqiu Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv:1909.09436*, 2019.

[24] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *EMNLP*, 2020.

[25] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*, 2019.

[26] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016.

[27] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, 2017.

[28] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023.

[29] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*, 2022.

[30] Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM TASLP*, 2023.

[31] Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In *ICML*, 2024.

[32] Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, and Mohammad Rastegari. Openelm: An efficient language model family with open training and inference framework. *arXiv:2404.14619*, 2024.

[33] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *ICLR*, 2017.

[34] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

[35] Hossein Mobahi, Mehrdad Farajtabar, and Peter L. Bartlett. Self-distillation amplifies regularization in hilbert space. In *NeurIPS*, 2020.

[36] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. In *NeurIPS*, 2024.

[37] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *NeurIPS*, 2019.

[38] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *NeurIPS*, 2020.

[39] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. In *ACL*, 2020.

[40] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *CHIL*, 2022.

[41] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019.

[42] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *NeurIPS*, 2024.

[43] Hao Peng, Xin Lv, Yushi Bai, Zijun Yao, Jiajie Zhang, Lei Hou, and Juanzi Li. Pre-training distillation for large language models: A design space exploration. In *ACL*, 2025.

[44] Tianyu Peng and Jiajun Zhang. Enhancing knowledge distillation of large language models through efficient multi-modal distribution alignment. In *COLING*, 2025.

[45] Phi3@Microsoft. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv:2404.14219*, 2024.

[46] Mary Phuong and Christoph H. Lampert. Towards understanding knowledge distillation. In *ICML*, 2019.

[47] Qwen. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

[48] Qwen. Qwen3 technical report. *arXiv:2505.09388*, 2025.

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

[50] Suhas Kamasetty Ramesh, Ayan Sengupta, and Tanmoy Chakraborty. On the generalization vs fidelity paradox in knowledge distillation. In *ACL (Findings)*, 2025.

[51] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[52] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*, 2020.

[53] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv:1910.01108*, 2020.

[54] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. In *EMNLP*, 2019.

[55] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *ICLR*, 2019.

[56] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *EMNLP*, 2019.

[57] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL*, 2019.

[58] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H. Chi, and Sagar Jain. Understanding and improving knowledge distillation. *arXiv:2002.03532*, 2021.

[59] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *NeurIPS*, 2020.

[60] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *ACL*, 2023.

[61] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *3rd Workshop on Noisy User-generated Text*, 2017.

[62] Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *EACL*, 2024.

[63] Xurong Xie, Zhucun Xue, Jiafu Wu, Jian Li, Yabiao Wang, Xiaobin Hu, Yong Liu, and Jiangning Zhang. Llm-oriented token-adaptive knowledge distillation. In *AAAI*, 2026.

[64] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 2020.

[65] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *CVPR*, 2020.

[66] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[67] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, 2019.

[68] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *ICCV*, 2019.

[69] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv:2401.02385*, 2024.

[70] Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. Revisiting knowledge distillation for autoregressive language models. In *ACL*, 2024.

## Appendix Contents


**A Detailed Related Works** **15**

**B Architecture Details** **15**

**C Evaluation Implementations** **16**

C.1 In-Domain Perplexity . . . . . 16

C.2 Out-of-Domain Perplexity . . . . . 16

C.3 Downstream Accuracy . . . . . 17

**D Pure Distillation** **17**

**E Choosing Best $\alpha$ Conditioning on All Evaluation Types** **18**

**F Training Curves** **19**

**G Additional Ablations** **20**

G.1 Generalization to Different Architecture Families . . . . . 20

G.2 Generalization to More Training Tokens . . . . . 21

**H Statistical Significance** **21**

**I Full Evaluation Results** **22**

**J Mechanism Study: Per-Token Analysis of Knowledge Distillation** **22**

J.1 Overview: PPL Improvement Across Alpha and Datasets . . . . . 22

J.2 Distillation Improves Hard Tokens . . . . . 23

J.3 Novel Information from the Teacher . . . . . 24

J.4 Distillation Is Not Label Smoothing . . . . . 24

J.5 Distribution Convergence: Weak Teachers Distort, Strong Teachers Transfer . . . . 25

J.6 Probability Redistribution, Not Sharpening . . . . . 25

J.7 Position Independence . . . . . 26

**K Limitations** **26**

# A Detailed Related Works

**Knowledge distillation.** Knowledge distillation (KD) [22] is a widely used technique to transfer knowledge from a teacher model to a student model by training the student to match the teacher's predictive distribution. The key insight is that the teacher's soft predictions contain "dark knowledge," information about the relative similarities between classes that is not present in hard labels [38, 58]. Originally proposed for model compression [6] in vision, KD has since been extended in many directions, including distilling intermediate representations [51], attention maps [66], and relational information between samples [41]. Various works have also studied the theoretical foundations of why distillation works, attributing its success to label smoothing effects [37, 64], regularization [35, 65], and improved optimization landscapes [12, 46, 4]. In this work, we focus on logit-level distillation for language models, where the student directly matches the teacher's output distribution.

**Distillation in language models.** While distillation was introduced in vision-based models, it already proves effective in large language models (LLMs). In language modeling, logit-level distillation is widely used to compress pretrained Transformers, from masked LMs [53, 24], to autoregressive LMs where distillation can be sensitive to the teacher and token-level behaviors [50]. Beyond matching output logits, prior work also distills intermediate signals to improve transfer [56, 59]. Distillation can further be realized through teacher-generated synthetic data, where the student is trained on sequences produced by the teacher [26, 60, 62]. While logit-level distillation is a common technique, most recent LLM distillation proceeds via teacher-generated synthetic data [1, 47, 48, 16]; in this work, we revisit logit-level distillation in pretraining and study when it improves performance.

**Design choices in LLM distillation.** Recent LLM distillation studies suggest that distillation success is governed by a small set of controllable factors rather than teacher strength alone. In pretraining, choices such as logits processing [63], loss design [19, 44], and generating teacher targets offline versus online [18, 43] can qualitatively change distillation outcomes. Moreover, KD is not monotonic in teacher strength: for autoregressive LMs, stronger teachers can sometimes degrade student performance [70, 8]. Prior work has shown that stronger teachers can sometimes degrade performance, but has not systematically characterized when this occurs in large-scale pretraining, where multiple components vary across a wide range. In this work, we focus on design choices related to the teacher side and conduct large-scale experiments varying all key components, such as architecture, tokens, and loss selection.

**Teacher-student relationships in distillation.** While knowledge distillation has long been viewed as a compression technique [6, 22], a growing body of work shows that this strong-to-weak framing is not necessary. In vision, Born-Again Networks [17] first showed that same-architecture distillation can still improve the student, with subsequent work clarifying why this effect holds [68, 12, 35]. In LLMs, while distillation remains a standard compression approach, strict strong-to-weak supervision may be unrealistic in the long run [7]. Consistent with this trend, industrial models could rely on prior-generation or cross-family models to supervise newer ones, exemplified in Qwen models [47, 48] and DeepSeek-R1 [16]. Despite these developments, systematic studies of how the scale of different components governs logit-level distillation in LLM pretraining remain limited. Busbridge et al. [8] study distillation scaling laws, but constrains teachers to a fixed budget distribution, while Peng et al. [43] studies how student size affects outcome but assumes access to a stronger pretrained teacher. To address these gaps, in this work we vary token budget and model size to study across diverse settings that result in all strong-to-weak, same-architecture, and weak-to-strong regimes.

# B Architecture Details

We use four model sizes based on the Llama3 architecture family. All models share the same architectural design choices: Grouped Query Attention (GQA) with a 4:1 ratio (32 query heads, 8 KV heads), SwiGLU activation (silu + linear gating), RMSNorm with $\epsilon = 10^{-5}$, RoPE positional encoding with max timescale 500,000, vocabulary size of 128,256, and no embedding tying. Table 4 details the configuration for each model size.

| 0.7B | | 1.7B | | 3.8B | | 8.0B | |
|---|---|---|---|---|---|---|---|
| Parameter | Value | Parameter | Value | Parameter | Value | Parameter | Value |
| Hidden dim | 1024 | Hidden dim | 2048 | Hidden dim | 3072 | Hidden dim | 4096 |
| Num layers | 12 | Num layers | 16 | Num layers | 28 | Num layers | 32 |
| MLP dim | 8192 | MLP dim | 8192 | MLP dim | 8192 | MLP dim | 14336 |
| Query heads | 32 | Query heads | 32 | Query heads | 32 | Query heads | 32 |
| KV heads | 8 | KV heads | 8 | KV heads | 8 | KV heads | 8 |
| Head dim | 128 | Head dim | 128 | Head dim | 128 | Head dim | 128 |
| Vocab size | 128,256 | Vocab size | 128,256 | Vocab size | 128,256 | Vocab size | 128,256 |
| RoPE base | 500,000 | RoPE base | 500,000 | RoPE base | 500,000 | RoPE base | 500,000 |
| Norm $\epsilon$ | $10^{-5}$ | Norm $\epsilon$ | $10^{-5}$ | Norm $\epsilon$ | $10^{-5}$ | Norm $\epsilon$ | $10^{-5}$ |

Table 4: **Architecture configurations for each model size.** All models share the same design choices (GQA, SwiGLU, RMSNorm, RoPE) but differ in hidden dimension, number of layers, and MLP dimension.

# C Evaluation Implementations

We evaluate models using three categories of metrics: in-domain perplexity, out-of-distribution perplexity, and downstream accuracy. This section details the implementation of each.

## C.1 In-Domain Perplexity

In-domain perplexity measures how well the model predicts text from the same distribution as the training data. We compute perplexity on a held-out test split of FineWeb-Edu [42] containing 1 billion tokens that are not seen during any model training (neither teacher pretraining nor student baseline or distillation).

Perplexity is defined as the exponentiated average cross-entropy loss (see Equation 4), where $N$ is the total number of tokens and $P(x_i \mid x_{<i})$ is the model's predicted probability of token $x_i$ given the preceding context. To compute perplexity, we tokenize the test set into a single long sequence, randomly split it into fixed-length chunks matching the model's context length (8192 tokens), run a forward pass to obtain logits, compute the cross-entropy loss between predicted logits and target tokens (shifted by one position), and average the loss across all tokens before exponentiating.

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(x_i \mid x_{<i})\right) \tag{4}$$

## C.2 Out-of-Domain Perplexity

Out-of-domain perplexity evaluates generalization to distributions beyond the training corpus. We measure perplexity on 11 curated corpora spanning diverse domains:

- **Web and general text:** Wikitext-103 [33] (Wikipedia articles), C4 [49] (Common Crawl web text)
- **Mathematics:** GSM8K [14] (grade school math problems), DM Mathematics [55] (mathematical text)
- **Code:** HumanEval [11] (code generation problems), CodeSearchNet [23] (code with documentation)
- **Scientific:** arXiv [15] (scientific paper abstracts), PubMedQA [25] (biomedical question-answer pairs)
- **News:** CNN-DailyMail [21] (news articles)
- **Legal:** ECHR [10] (European Court of Human Rights cases)
- **Multilingual:** XQuAD [3] (multilingual question-answering across 11 languages)

Each corpus is formatted according to its structure. For plain text corpora (Wikitext, arXiv), documents are joined with double newlines. For QA datasets (GSM8K, PubMedQA), question-answer pairs are concatenated. For code (HumanEval), prompts and canonical solutions are combined. Perplexity is computed using the same procedure as in-domain evaluation.

### C.3 Downstream Accuracy

Downstream accuracy evaluates transfer to tasks requiring capabilities beyond next-token prediction. We use 15 multiple-choice benchmarks covering broad knowledge, reasoning, and comprehension.

**Scoring mechanism.** For multiple-choice tasks, the model does not generate text. Instead, we use log-likelihood scoring: for each question, we construct prompt-answer pairs for each possible choice, compute the log-likelihood $\log P(\text{answer} \mid \text{question})$ for each choice, and select the answer with the highest log-likelihood as the prediction.

For a context-continuation pair, we tokenize both, concatenate them, run a forward pass to obtain logits, apply log-softmax, and sum the log probabilities of the continuation tokens (see Equation 5). Some benchmarks use length-normalized accuracy to prevent bias toward shorter answers. In this case, the log-likelihood is divided by the number of characters in the continuation before comparison.

$$\log P(\text{continuation} \mid \text{context}) = \sum_{j=1}^{|\text{cont}|} \log P(t_j \mid \text{context}, t_{<j}) \tag{5}$$

**Benchmarks.** We evaluate on 15 benchmarks with the following configurations:

- **Broad knowledge:** MMLU [20] (57 academic subjects, 5-shot, 4 choices)
- **Science and math:** ARC-Easy [13] (elementary science, 0-shot, 3-5 choices), SciQ [61] (science QA, 0-shot, 4 choices), OpenBookQA [34] (science facts, 5-shot, 4 choices), MathQA [2] (math word problems, 5-shot, 5 choices)
- **Commonsense reasoning:** HellaSwag [67] (sentence completion, 0-shot, 4 choices), PIQA [5] (physical intuition, 0-shot, 2 choices), WinoGrande [52] (coreference, 5-shot, 2 choices), Social IQa [54] (social commonsense, 0-shot, 3 choices), CommonsenseQA [57] (general commonsense, 10-shot, 5 choices)
- **Reading comprehension:** RACE [27] (English exams, 0-shot, 4 choices)
- **Logical reasoning:** LogiQA 2.0 [30] (logical reasoning, 10-shot, 4 choices)
- **Truthfulness:** TruthfulQA MC1 [29] (identifying truthful answers among misconceptions, 0-shot with 10 in-prompt examples, variable choices)
- **Natural language inference:** ANLI R1 [39] (adversarial entailment, 10-shot, 3 choices)
- **Medical:** MedMCQA [40] (medical entrance exams, 10-shot, 4 choices)

**Length normalization and few-shot selection.** Four benchmarks use length-normalized accuracy to prevent bias toward shorter answers: HellaSwag, ARC-Easy, PIQA, and OpenBookQA. These benchmarks score the full text of answer choices, which vary significantly in length. The remaining benchmarks use raw accuracy, as their answer choices are either single letters or similar-length phrases. For few-shot evaluation, we use deterministic selection: examples are drawn from the first $N$ samples of the training split (or dev split for MMLU). This ensures reproducibility across runs.

## D Pure Distillation

Pure distillation sets the loss mixing coefficient $\alpha = 1.0$, using only the knowledge distillation loss without the language modeling loss. This is a common practice in many implementations, as it simplifies training by removing the need to tune $\alpha$.

Table 5 shows distillation improvement under pure distillation. The results reveal a clear divide based on teacher strength. For strong teachers (large architecture with high token budgets), pure distillation yields substantial improvement across all evaluation types. For example, the 8.0B teacher at 80B tokens achieves +3.7% in-domain perplexity, +7.0% out-of-distribution perplexity, and +3.9% downstream accuracy. However, for weak teachers, pure distillation degrades performance significantly. The 0.7B teacher at 10B tokens yields -68.7% in-domain perplexity and -146.5% out-of-distribution perplexity (number can exceed 100% because $ppl \in [1, \infty)$), indicating that relying entirely on a weak teacher's signal without ground-truth supervision is harmful. Additionally, we also observe the same trend from our mixed loss experiments: stronger teachers do not always

| | %Δ in-domain ppl (↑) | | | | avg. %Δ out-of-distribution ppl (↑) | | | | avg. %Δ downstream acc (↑) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B |
| 10B | -68.7% | -36.1% | -20.6% | **-11.2%** | -146.5% | -65.1% | -42.0% | **-24.6%** | -8.6% | -4.9% | -2.5% | **-0.4%** |
| 30B | -31.6% | -11.5% | -2.2% | **1.4%** | -44.3% | -13.8% | -3.0% | **0.8%** | -4.5% | -2.4% | **+1.1%** | +1.0% |
| 50B | -21.8% | -5.9% | 0.9% | **3.0%** | -28.5% | -5.6% | 2.9% | **5.7%** | -3.3% | -0.9% | **+1.9%** | +1.4% |
| 80B | -15.8% | -2.6% | 2.4% | **3.7%** | -16.1% | 0.9% | 5.8% | **7.0%** | -1.8% | +1.4% | +1.4% | **+3.9%** |
| 100B | -13.8% | -1.5% | 2.8% | **3.8%** | -12.2% | 1.6% | 6.3% | **7.3%** | -3.3% | +1.1% | +1.2% | **+3.2%** |
| 300B | -7.0% | 1.6% | 3.6% | **3.9%** | -5.2% | 4.9% | **6.9%** | 6.8% | -0.3% | **+3.6%** | +2.1% | +1.2% |

Table 5: **Distillation improvement under pure distillation ($\alpha = 1.0$).** Rows indicate teacher token budget; columns indicate teacher architecture size. Pure distillation works for strong teachers (large architecture, high token budget) but degrades significantly for weak teachers, particularly smaller architectures with fewer training tokens.

| | %Δ in-domain ppl (↑) | | | | avg. %Δ out-of-distribution ppl (↑) | | | | avg. %Δ downstream acc (↑) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B |
| 10B | -65.8% | -34.7% | -20.2% | **-11.5%** | -143.3% | -64.0% | -43.2% | **-26.2%** | -8.1% | -4.5% | -1.7% | **-1.3%** |
| 30B | -30.4% | -12.0% | -4.1% | **-1.8%** | -43.8% | -15.3% | -7.4% | **-4.6%** | -5.8% | -3.8% | **-1.5%** | -3.3% |
| 50B | -21.5% | -6.9% | -2.1% | **-0.9%** | -30.1% | -9.1% | -3.2% | **-1.6%** | -4.0% | -2.4% | **-1.5%** | -1.8% |
| 80B | -15.9% | -4.6% | -1.2% | **-0.6%** | -18.7% | -4.4% | -1.9% | **-0.6%** | -2.7% | -0.4% | -2.2% | **0.0%** |
| 100B | -14.2% | -3.8% | -1.1% | **-0.4%** | -15.3% | -4.4% | **-1.3%** | -1.6% | -4.0% | -0.8% | -2.1% | **0.0%** |
| 300B | -7.9% | -1.7% | -0.7% | **-0.1%** | -9.6% | -2.7% | -0.5% | **-0.4%** | -1.8% | **-0.7%** | -1.4% | -1.6% |

Table 6: **Difference of effectiveness between pure distillation and optimal $\alpha$ distillation.** Negative values indicate pure distillation underperforms optimal $\alpha$ selection. The gap is largest for weak teachers, where pure distillation can be over 60% worse than optimal $\alpha$ on in-domain perplexity. For strong teachers (8.0B at high token budgets), the gap narrows, but pure distillation is still generally worse, indicating that pure distillation approaches optimal performance when the teacher is sufficiently strong.

yield better students. For downstream accuracy, the 8.0B teacher peaks at 80B tokens (+3.9%) and declines with further training (+3.2% at 100B, +1.2% at 300B).

Table 6 compares pure distillation to optimal $\alpha$ selection. In almost all configurations, there exists a better loss mixing than pure distillation. The gap is substantial for weak teachers: pure distillation underperforms optimal $\alpha$ by over 60% on in-domain perplexity for small teachers with few training tokens. Even for strong teachers, pure distillation rarely matches optimal $\alpha$. Only a few configurations with the strongest teachers (8.0B at high token budgets) show near-zero difference, where pure distillation happens to be optimal. These findings suggest that tuning $\alpha$ is worthwhile across most teacher configurations.

## E Choosing Best $\alpha$ Conditioning on All Evaluation Types

In the main experiments (§4), we select the optimal loss mixing coefficient $\alpha$ independently for each evaluation type, allowing different $\alpha$ values for in-domain perplexity, out-of-distribution perplexity, and downstream accuracy. Here we present results using a single $\alpha$ per teacher configuration, selected to maximize average improvement across all benchmarks.

**Methodology.** For each teacher configuration, we compute the average percentage improvement across all 27 benchmarks (1 in-domain perplexity + 11 out-of-distribution perplexity corpora + 15 downstream accuracy tasks), weighting each benchmark equally. We then select the $\alpha$ that maximizes

| | %Δ in-domain ppl (↑) | | | | avg. %Δ out-of-distribution ppl (↑) | | | | avg. %Δ downstream acc (↑) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B | 0.7B | 1.7B | 3.8B | 8.0B |
| 10B | -2.9% (0.2) | -1.4% (0.2) | **-0.4%** (0.2) | -0.5% (0.4) | -3.2% (0.2) | -1.1% (0.2) | 1.2% (0.2) | **1.6%** (0.4) | -0.5% (0.2) | -0.4% (0.2) | -1.0% (0.2) | **0.9%** (0.4) |
| 30B | -1.2% (0.2) | 0.5% (0.2) | 1.9% (0.5) | **3.2%** (0.6) | -0.5% (0.2) | 1.5% (0.2) | 4.4% (0.5) | **4.7%** (0.6) | 1.3% (0.2) | 1.4% (0.2) | 2.6% (0.5) | **4.3%** (0.6) |
| 50B | -0.3% (0.2) | 1.0% (0.2) | 3.0% (0.6) | **3.9%** (0.8) | 1.6% (0.2) | 3.0% (0.2) | 6.0% (0.6) | **7.3%** (0.8) | 0.7% (0.2) | 1.5% (0.2) | **3.4%** (0.6) | 3.2% (0.8) |
| 80B | -1.3% (0.4) | 1.6% (0.6) | 3.5% (0.8) | **3.7%** (1.0) | 2.6% (0.4) | 5.3% (0.6) | **7.7%** (0.8) | 7.0% (1.0) | 0.9% (0.4) | 0.9% (0.6) | 3.6% (0.8) | **3.9%** (1.0) |
| 100B | 0.4% (0.2) | 2.3% (0.5) | **3.8%** (0.6) | 3.8% (1.0) | 3.1% (0.2) | 5.1% (0.5) | 7.0% (0.6) | **7.3%** (1.0) | 0.1% (0.2) | 1.9% (0.5) | **3.3%** (0.6) | 3.2% (1.0) |
| 300B | -0.3% (0.6) | 3.3% (0.6) | **4.3%** (0.8) | 4.0% (0.8) | 4.4% (0.6) | **7.6%** (0.6) | 7.4% (0.8) | 6.6% (0.8) | 1.5% (0.6) | **3.2%** (0.6) | 2.8% (0.8) | 2.8% (0.8) |

Table 7: **Distillation improvement under jointly optimized $\alpha$ across all benchmarks.** The student is 1.7B trained on 50B tokens. For each teacher configuration, a single $\alpha$ is selected to maximize average improvement across all 27 benchmarks. Results closely match those from per-metric optimization (Table 2), confirming that our findings are robust to $\alpha$ selection methodology.

this average. This approach naturally weights toward out-of-distribution and downstream metrics, which comprise 26 of the 27 benchmarks.

**Results.** Table 7 shows the results under jointly optimized $\alpha$. The patterns closely match those from per-metric optimization in Table 2: weak-to-strong distillation remains effective, same-level distillation yields improvement, and stronger teachers do not always produce better students. The jointly optimal $\alpha$ values follow the same trend as per-metric values, where weaker teachers favor lower $\alpha$, while stronger teachers favor higher $\alpha$. The slight differences in improvement magnitude (compared to Table 2) reflect the trade-off from using a single $\alpha$ across metrics with potentially different optima. These results confirm that our findings are robust to the $\alpha$ selection methodology.

## F Training Curves

Figure 6 shows training loss curves for distillation pretraining across all teacher configurations. Each subplot corresponds to a teacher configuration, with different colors indicating the loss mixing coefficient $\alpha$. The distillation training runs converge steadily. Across all configurations, higher $\alpha$ leads to faster loss reduction in the initial training stage, as the student receives stronger supervision from the teacher's distribution. However, this initial advantage does not always translate to lower final loss: for weak teachers, high $\alpha$ eventually converges to higher loss than low $\alpha$, while for strong teachers, high $\alpha$ maintains its advantage throughout.

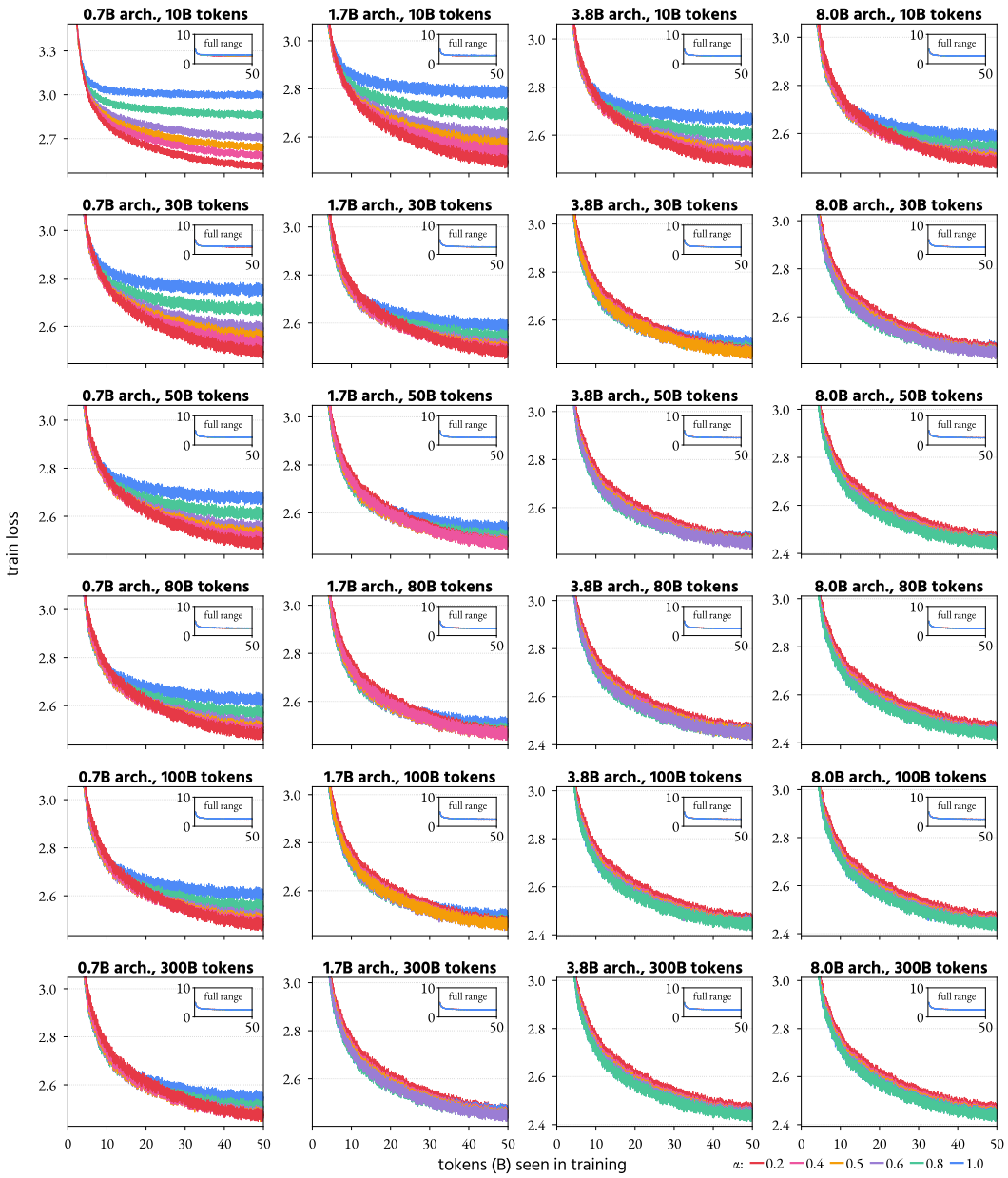


Figure 6: Training loss curves for distillation pretraining across all teacher configurations. Columns are ordered by teacher architecture size and rows by teacher token budget. Each subplot shows training loss versus tokens seen by the 1.7B student, zoomed in for clarity, with different colors indicating loss mixing coefficient $\alpha$. Insets show the full y-axis range for reference. Across all configurations, higher $\alpha$ leads to faster loss reduction in the initial training stage, regardless of whether it eventually converges to the lowest or highest final loss among the $\alpha$ values.

# G Additional Ablations

## G.1 Generalization to Different Architecture Families

Our main experiments use the Llama3 architecture family. We now test whether our findings transfer to a different architecture by using Qwen3 [48], with variants of 0.4B, 0.5B, 0.8B, and 1.2B parameters. We use the 0.5B model as the student and train teachers, student, and baseline for 10B

| | 0.4B | 0.5B | 0.8B | 1.2B |
|---|---|---|---|---|
| in-domain ppl | -2.5% ($\alpha$=0.2) | -0.1% ($\alpha$=0.2) | +2.1% ($\alpha$=0.5) | +3.2% ($\alpha$=0.8) |
| ood ppl | +0.2% ($\alpha$=0.2) | +1.6% ($\alpha$=0.4) | +6.5% ($\alpha$=0.5) | +6.1% ($\alpha$=0.6) |
| downstream acc | +0.1% ($\alpha$=0.2) | +0.2% ($\alpha$=0.2) | +1.0% ($\alpha$=0.2) | +0.7% ($\alpha$=0.4) |

Table 8: **Distillation improvement with Qwen3 architecture.** Student is 0.5B trained on 10B tokens. Results show percentage improvement over standard baseline.

| | 0.7B | 1.7B | 3.8B | 8.0B |
|---|---|---|---|---|
| in-domain ppl | -0.8% | +0.2% | +0.9% | +0.9% |
| ood ppl | +2.4% | +2.8% | +2.4% | +3.4% |
| downstream acc | +0.3% | +1.0% | +3.4% | +3.1% |

Table 9: **Distillation improvement at 300B training tokens with $\alpha = 0.2$.** Student is 1.7B trained on 300B tokens. Results show percentage improvement over standard pretraining 300B baseline.

tokens, sweeping $\alpha$ values. The results in Table 8 again confirm our main findings: weak-to-strong distillation works, and larger teachers do not always yield better distillation.

### G.2 Generalization to More Training Tokens

Our main experiments train students on 50B tokens. We now test whether our findings hold at larger scale by training with 300B tokens for teacher, student, and baseline. We evaluate only $\alpha = 0.2$ as it is easiest for improvement. The results in Table 9 confirm our main findings: weak-to-strong distillation remains effective, and at this $\alpha$, the 3.8B teacher outperforms the 8.0B teacher in downstream performance.

## H Statistical Significance

To verify that reported improvements are not driven by seed-to-seed variation, we measure baseline variance across multiple training seeds for each architecture size. We retrain the standard pretraining baselines at all four model sizes (0.7B, 1.7B, 3.8B, 8.0B at 50B tokens) and report mean and standard deviation of the resulting in-domain perplexity, average out-of-distribution perplexity, and average downstream accuracy.

| | in-domain ppl | | ood ppl avg. | | downstream acc | |
|---|---|---|---|---|---|---|
| Model | mean | SD | mean | SD | mean | SD |
| 0.7B | 14.9 | 0.051 | 20.6 | 1.04 | 0.381 | 0.0014 |
| 1.7B | 12.2 | 0.028 | 15.7 | 1.06 | 0.400 | 0.0015 |
| 3.8B | 10.3 | 0.058 | 13.0 | 1.22 | 0.426 | 0.0098 |
| 8.0B | 9.1 | 0.069 | 11.7 | 1.05 | 0.436 | 0.0003 |

Table 10: **Baseline variance across multiple training seeds.** For each architecture size, mean and standard deviation are computed across multi-seed standard pretraining runs at 50B tokens. In-domain perplexity standard deviation is below 0.07 (less than 0.5% of the mean) and downstream accuracy standard deviation is below 0.01 across all sizes. Out-of-distribution perplexity exhibits larger absolute variance because it averages 11 corpora with different scales, but its relative magnitude remains around 5–10% of the mean.

In-domain perplexity standard deviation stays below 0.5% of the mean for every architecture size, and downstream accuracy standard deviation stays below 0.01. Reported distillation improvements throughout the paper (e.g., +3.9% in-domain perplexity for the 3.8B teacher, up to +8.9% out-of-distribution perplexity for the 8.0B teacher, +4.3% downstream accuracy for the 1.7B teacher) exceed these seed-noise floors by an order of magnitude. Combined with the monotonic structures observed across teacher size, training tokens, and the loss mixing coefficient $\alpha$ (§4, §5), the reported patterns reflect genuine distillation effects rather than random variation.

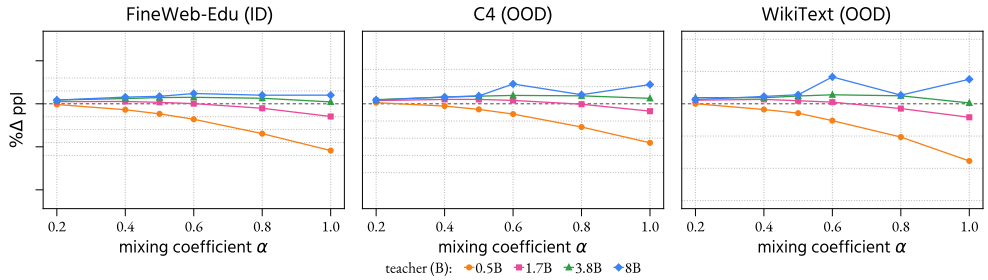


Figure 7: **Percentage PPL improvement over baseline across alpha values and datasets.** Each panel is a dataset (FineWeb-Edu in-domain, C4 and WikiText out-of-distribution). Lines represent teacher sizes; the dashed line at $y$=0 is the baseline. Strong teachers improve the student across all alpha values; weak teachers degrade performance at high alpha. Optimal alpha increases with teacher strength.

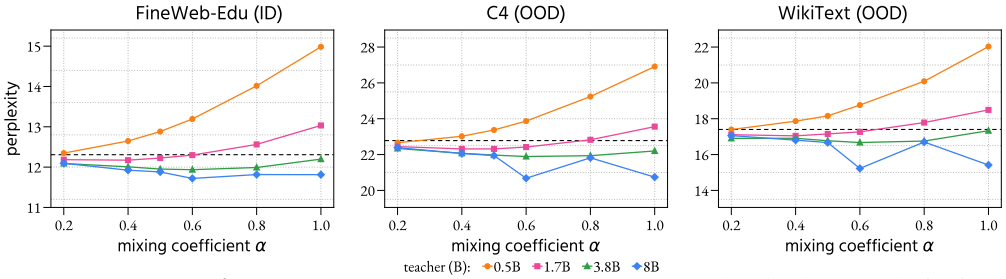


Figure 8: **Raw perplexity values across alpha values and datasets.** The absolute perplexity instead of relative improvement. Dashed lines indicate the baseline. Lines below the dashed line indicate better-than-baseline performance.

# I Full Evaluation Results

We report full per-benchmark evaluation results for all teacher-student configurations. Table 11 presents perplexity results across 12 benchmarks, including in-domain and out-of-distribution evaluations. Table 12 presents downstream accuracy results across 15 benchmarks. For each teacher configuration, we report the best-performing loss mixing coefficient $\alpha$ along with the corresponding improvement over the baseline student.

# J Mechanism Study: Per-Token Analysis of Knowledge Distillation

To understand *how* distillation transfers knowledge, we conduct a per-token mechanism study. We collect per-token logit data ($\sim$500K tokens per model) on three datasets: FineWeb-Edu (in-domain), C4 (out-of-distribution), and WikiText (out-of-distribution). We analyze 29 models: a 1.7B baseline (standard pretraining), 4 teachers (0.5B, 1.7B, 3.8B, 8.0B, all trained on 50B tokens), and 24 distilled 1.7B students (4 teachers $\times$ 6 alpha values: 0.2, 0.4, 0.5, 0.6, 0.8, 1.0).

## J.1 Overview: PPL Improvement Across Alpha and Datasets

Figure 7 shows percentage PPL improvement over the baseline as alpha varies. Strong teachers (8.0B, 3.8B) consistently improve the student across all datasets, with 8.0B achieving up to $\sim$9% improvement on C4. The 1.7B same-size teacher provides modest improvement ($\sim$2%), peaking around $\alpha$=0.4. The 0.5B teacher hurts at all $\alpha > 0.2$, with degradation increasing sharply at higher alpha (up to $-18\%$ on C4 at $\alpha$=1.0). Optimal alpha increases with teacher quality: $\sim$0.2 for 0.5B, $\sim$0.4 for 1.7B, and $\sim$0.6 for 3.8B/8.0B. These patterns are consistent across in-domain and out-of-distribution datasets. Figure 8 shows the corresponding raw perplexity.

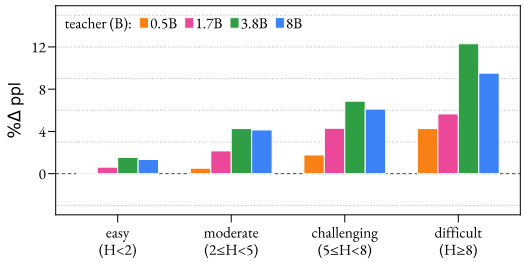


Figure 9: **Improvement by token difficulty at each teacher's best alpha.** Token difficulty bins are defined by the baseline model's entropy. Improvement is concentrated on harder tokens for all, refuting the regularization hypothesis. Stronger teachers show larger improvement at every level, but the monotonic concentration on hard tokens is universal.




Figure 10: **PPL improvement by token difficulty across all alpha values.** Each panel corresponds to a teacher size. Rows are alpha values (0.2 to 1.0); columns are difficulty bins. Green indicates improvement; red indicates degradation. For weak teachers, degradation spreads from easy to hard tokens as alpha increases. For strong teachers, improvement is robust across all alpha values, with the strongest gains consistently on difficult tokens.

### J.2 Distillation Improves Hard Tokens

A central question is whether distillation acts as a form of regularization (benefiting tokens uniformly) or as genuine knowledge transfer (benefiting tokens where the baseline is most uncertain). We test this by binning tokens into four difficulty levels based on the baseline model's entropy $H$: easy ($H < 2$), moderate ($2 \leq H < 5$), hard ($5 \leq H < 8$), and difficult ($H \geq 8$).

Figure 9 shows PPL improvement by difficulty bin at each teacher's best alpha. Improvement is monotonically concentrated on harder tokens for *all* teachers. Easy tokens see ∼0–1% improvement while difficult tokens see ∼4–12% improvement. This pattern directly refutes the regularization hypothesis: regularization such as label smoothing would produce uniform improvement across difficulty levels. Instead, distillation specifically helps where the baseline is most uncertain. Stronger teachers provide more improvement at every difficulty level, but the concentration on hard tokens is universal, even the 0.5B teacher at its best $\alpha$=0.2 shows the same pattern at smaller magnitude.

Figure 10 provides the complete picture across all 24 students, not just best alpha. For the 0.5B teacher, low alpha yields mild improvement on hard tokens, but as alpha increases, easy and moderate tokens degrade first, followed by hard tokens. At $\alpha$=1.0, all bins are deeply negative ($-9.5\%$ to $-26.3\%$). For the 3.8B and 8.0B teachers, virtually all cells are positive, with the difficult-token column consistently showing the strongest improvement (up to +13.3% for 8.0B at $\alpha$=0.4–0.8).

Figure 11 confirms statistical significance via bootstrap confidence intervals. We define hard-token concentration as the improvement gap between hard and easy tokens. For all students with positive overall improvement, 100% show statistically significant hard-token concentration ($p < 0.001$). Non-significant cases are confined to edge cases where overall improvement is near zero or negative (0.5B at $\alpha \geq 0.5$, 1.7B at $\alpha$=1.0). When distillation hurts (0.5B at $\alpha$=0.8, 1.0), the concentration goes strongly negative: hard tokens are damaged *more* than easy tokens, consistent with a weak teacher actively misleading student's most uncertain predictions.

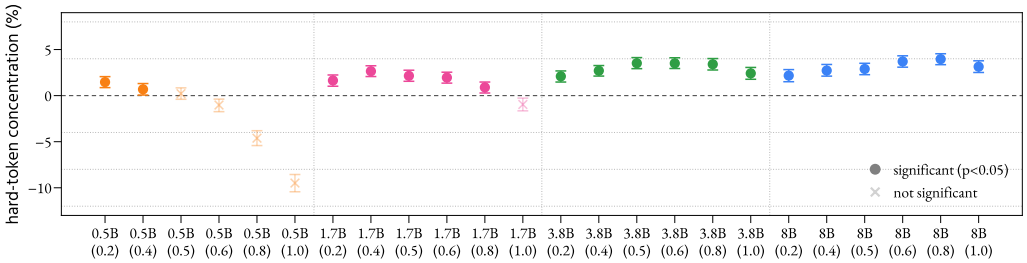


Figure 11: **Statistical significance of hard-token concentration.** Hard-token concentration is the improvement gap between hard and easy tokens. Filled circles indicate statistical significance ($p < 0.05$, bootstrap with 1000 resamples); faded crosses indicate non-significance. Error bars show 95% confidence intervals. All students with positive overall improvement show significant hard-token concentration. When distillation hurts (0.5B at high alpha), concentration reverses, where hard tokens are damaged more than easy tokens.

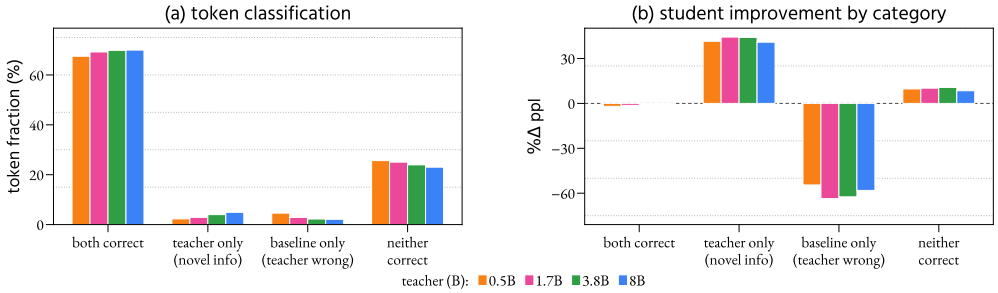


Figure 12: **Causal analysis of distillation by token category.** Tokens are classified by whether the ground-truth appears in the top-10 predictions of the baseline and/or teacher. Panel (a): fraction of tokens in each category. Panel (b): student PPL improvement per category. Improvement comes from "teacher only" tokens (+30–40%) where the teacher provides novel correct predictions. "Baseline only" tokens ($-55\%$ to $-65\%$) represent the cost of distillation. This explains why $\alpha < 1$ is optimal: it limits damage from tokens where the teacher is wrong but the baseline is right.

### J.3 Novel Information from the Teacher

To identify the causal mechanism behind distillation's improvement, we classify each token into four categories based on whether the ground-truth token appears in the top-10 predictions of the baseline and teacher: (1) "both correct" ($\sim$68% of tokens), (2) "teacher only (novel info)" ($\sim$2–5%), (3) "baseline only (teacher wrong)" ($\sim$2–4%), and (4) "neither correct" ($\sim$23%).

Figure 12 reveals the mechanism. On "both correct" tokens, the student shows $\sim$0% improvement — when both models already know the answer, the teacher adds nothing. On "teacher only" tokens, improvement is massive (+30–40%) — the teacher provides novel correct information that the baseline lacks, and the student absorbs it. On "baseline only" tokens, the teacher actively misleads the student ($-55\%$ to $-65\%$) — this is the cost of distillation. On "neither correct" tokens, the student still improves ($\sim$+10%), suggesting the teacher's soft distribution is informative even when neither model's top predictions are correct.

These results explain two key observations. First, stronger teachers help more because they have more "teacher only" tokens (8.0B: 4.9% vs. 0.5B: 2.3%). Second, $\alpha < 1$ is optimal because it limits the damage from the "baseline only" category where the teacher's signal is harmful.

### J.4 Distillation Is Not Label Smoothing

A common hypothesis is that distillation acts similarly to label smoothing, a uniform regularizer. Figure 13 directly tests this by comparing the token-level benefit profiles of label smoothing and distillation. Label smoothing's benefit is analytically proportional to $\log V - H$, where $V$ is the vocabulary size and $H$ is the token entropy, meaning it helps most on easy tokens where the model

is overconfident. Distillation's measured benefit shows the opposite pattern: it helps most on hard tokens.

The two profiles cross: label smoothing benefit decreases with token difficulty while distillation benefit increases. This definitively refutes the hypothesis that distillation is merely a form of regularization similar to label smoothing; they operate on complementary parts of the token difficulty spectrum.

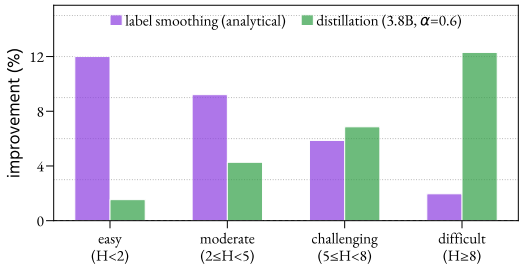


Figure 13: **Distillation vs. label smoothing: opposite benefit profiles.** Purple bars show the analytical label smoothing benefit profile (proportional to $\log V - H$); green bars show the measured distillation improvement (3.8B teacher, $\alpha$=0.6). Label smoothing helps most on easy tokens; distillation helps most on hard tokens. The crossing pattern refutes the hypothesis that distillation is a form of regularization.

### J.5 Distribution Convergence: Weak Teachers Distort, Strong Teachers Transfer

We measure whether the distilled student's output distribution moves toward the teacher or stays close to the baseline by computing a convergence ratio: the top-10 token overlap between student and teacher, divided by the overlap between the distillation pretraining student and standard pretraining baseline. A ratio of 1 means equidistant; ratio $> 1$ means the student is closer to the teacher.

Figure 14 shows that the 0.5B teacher distorts the student distribution: the ratio rises from 0.96 to 1.12 as alpha increases, meaning the student's predictions increasingly resemble the weaker teacher's. In contrast, the 3.8B teacher ($\sim$0.97–1.01) and 8.0B teacher ($\sim$0.95–0.97) transfer knowledge without distorting the student's distribution. This mechanistically explains why optimal alpha is lower for weak teachers: at high alpha, weak teachers pull the student away from its natural distribution, causing the performance degradation.

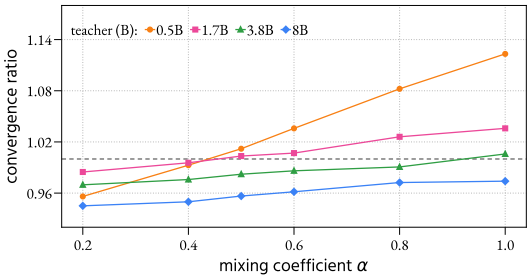


Figure 14: **Distribution convergence ratio across alpha.** The ratio measures top-10 token overlap (student–teacher) divided by overlap (student–baseline). Ratio $> 1$ means the student is closer to the teacher; ratio $< 1$ means closer to the baseline. The 0.5B teacher distorts the student distribution at high alpha (ratio up to 1.12), while the 3.8B and 8.0B teachers transfer knowledge without distortion (ratio $\leq 1.01$). This explains why weak teachers require low alpha.

### J.6 Probability Redistribution, Not Sharpening

A natural expectation is that distillation sharpens predictions (lowering entropy) to improve perplexity. Figure 15 tests this by plotting each student's entropy change against its PPL improvement. Most successful students have *positive* entropy increase (less confident) yet *positive* PPL improvement (better predictions). This counterintuitive result indicates that distillation redistributes probability mass from wrong confident predictions to better-calibrated ones, rather than simply sharpening.

Strong teachers (8.0B, 3.8B) cluster in the upper region, modest entropy change with strong PPL improvement, indicating efficient knowledge transfer. The 0.5B teacher at high alpha occupies the

lower-right region, large entropy increase *and* PPL degradation, the weak teacher makes the student both less confident and accurate.

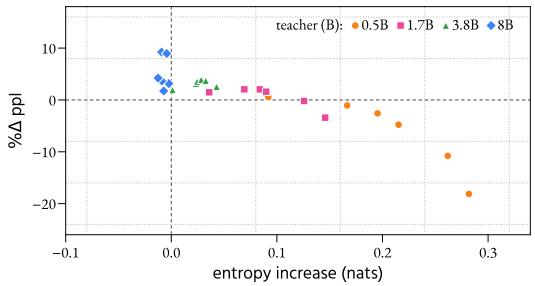


Figure 15: **Entropy change vs. PPL improvement.** Each point is one distilled student. Most successful students show increased entropy (less confident) yet improved PPL (more accurate), indicating distillation redistributes probability rather than sharpening predictions. Strong teachers achieve high improvement with modest entropy change; the 0.5B teacher at high alpha increases entropy substantially while degrading PPL.

### J.7 Position Independence

Figure 16 tests whether distillation improvement varies with position in the context window. PPL improvement is computed in four position bins (0–128, 128–512, 512–1024, 1024–2048 tokens). All teachers show essentially flat improvement across positions, there is no early-token vs. late-token effect. This rules out the hypothesis that distillation primarily helps with "cold start" predictions (early tokens with little context) or with long-range dependencies (late tokens). The uniformity supports that distillation transfers token-level knowledge about *what* to predict given the context, rather than position-dependent structural knowledge.

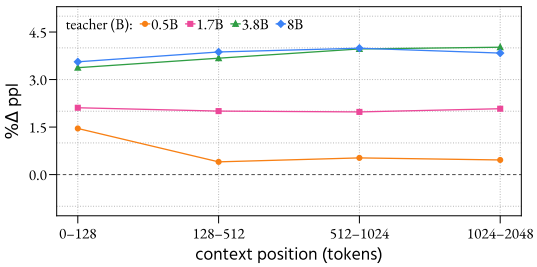


Figure 16: **PPL improvement by context position.** Each line represents a teacher at its best alpha, with improvement measured in four position bins across the context window. Improvement is uniform across positions for all teachers, ruling out position-dependent effects and supporting that distillation transfers token-level predictive knowledge.

## K Limitations

While our study covers a broad sweep of teacher-student configurations, several limitations remain.

**Single-seed main-table results.** Our study comprises over 150 full pretraining runs and consumed roughly 200,000 H100-equivalent GPU hours, so running additional seeds for every cell of the 24-teacher $\times$ 6-$\alpha$ grid was infeasible within the budget that was available to us. We do report bootstrap confidence intervals for the per-token mechanism study (Figure 11), and the breadth of the sweep itself functions as a robustness check: the qualitative trends are required to hold simultaneously across teacher size, teacher family, training-token count, and mixing weight, which is far harder for seed noise to produce than any single comparison would be. Random noise cannot produce such patterns across 24 configs $\times$ 6 alphas $\times$ 25 benchmarks. We therefore expect numerical shifts under reseeding to be small relative to the effect sizes we report, even though we cannot directly bound them.

**Logit-level distillation only.** We study forward-KL distillation on next-token distributions and do not evaluate hidden-state / feature-level distillation, attention-map distillation, or synthetic-data (teacher-generated text) approaches. We made this choice because logit-level KD is the dominant form of distillation used in modern open-weight pretraining recipes (e.g. Gemma 2/3, Minitron) and is the only form that does not require access to the teacher's training data or sampling pipeline, making it the most directly actionable setting for practitioners. It also exercises the teacher signal on every training token, which is the strongest possible test of whether additional teacher capability translates into student gains; conclusions in this regime should still be informative even when the exact numbers will differ for other transfer mechanisms.

**Tokenizer-compatible architecture families.** Logit-level KD requires the teacher and student to share a vocabulary, so we restrict experiments to the Llama3 and Qwen3 families and do not test cross-family transfer (e.g. Llama-teacher with a custom-tokenizer student) or vocabulary-projection methods. We chose these two families because they are among the most widely deployed open model lines and span meaningfully different pretraining data mixes and training recipes; the fact that the same teacher-strength trend appears in both, rather than only in one, suggests the effect is a property of the KD signal itself rather than of any one provider's recipe.

**Scale ceiling.** Teachers range up to 8.0B parameters and 300B tokens, and the student is 1.7B parameters trained on 50B tokens, so we cannot directly test extrapolation to frontier-scale students (tens or hundreds of billions of parameters) or to much longer training horizons. This ceiling is set by the same compute envelope that motivates our single-seed grid; pushing it up by an order of magnitude would multiply the cost of every cell in the sweep. Within the range we do cover, the trends are consistent across the student sizes in §6, which is evidence that the conclusion is governed by the structure of the signal rather than by a particular operating point on the scaling curve.

**Pretraining-only setting.** We study distillation during the pretraining phase and do not measure how the choice of teacher interacts with later stages such as supervised fine-tuning, instruction tuning, or RLHF. We focus here because pretraining is where the cost asymmetry between teacher serving and student training is most extreme and where current prescriptive guidance most strongly advocates the strongest available teacher, so it is the setting in which our finding has the largest immediate leverage. It is also the stage that determines the representations all later stages build on, meaning the teacher choice studied here plausibly has the longest-lived downstream effect even if the optimal teacher for post-training distillation differs.

**Evaluation coverage.** Our 11 OOD perplexity corpora and 15 downstream benchmarks are predominantly English, and we do not evaluate multilingual transfer, specialized code benchmarks beyond standard ones, or long-context capabilities. The compute budget that paid for the experimental sweep was deliberately not split into expanding the evaluation surface, on the reasoning that breadth across teachers and $\alpha$ values yields more signal about the central question than breadth across evaluations would. Within the surface we do cover, which spans commonsense, knowledge, reasoning, and reading comprehension, the qualitative finding reproduces on every subset rather than being driven by any single benchmark.

Table 11: **Full perplexity evaluation results.** We include (1) in-domain perplexity evaluation on FineWeb-Edu (FW); (2) out-of-distribution perplexity evaluation on Wikitext-103 (Wiki), C4 (C4), GSM8K (GSM), DM Mathematics (DMM), HumanEval (HEval), CodeSearchNet (CSN), arXiv (AX), CNN-DailyMail (CNN), ECHR (ECHR), PubMedQA (PM), and XQuAD (XQA).

| perplexity (↓) | FW | Wiki | C4 | GSM | DMM | HEval | CSN | AX | CNN | ECHR | PM | XQA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| → *standard pretrained student baseline* | | | | | | | | | | | | |
| | 12.17 | 16.01 | 21.53 | 6.61 | 9.33 | 13.54 | 20.79 | 24.48 | 14.27 | 9.24 | 24.77 | 12.13 |
| → *0.7B arch. 10B tokens teacher config ($\alpha$=0.2)* | | | | | | | | | | | | |
| pretrained teacher | 21.21 | 32.96 | 36.50 | 16.94 | 14.44 | 44.51 | 79.14 | 47.53 | 25.72 | 15.10 | 50.07 | 74.66 |
| distilled student | 12.52 | 16.49 | 21.92 | 6.91 | 9.15 | 14.06 | 20.70 | 25.33 | 14.70 | 9.44 | 24.88 | 14.02 |
| improvement (%) | -2.9 | -3.0 | -1.8 | -4.5 | +1.9 | -3.8 | +0.4 | -3.5 | -3.1 | -2.1 | -0.5 | -15.6 |
| → *0.7B arch. 30B tokens teacher config ($\alpha$=0.2)* | | | | | | | | | | | | |
| pretrained teacher | 16.25 | 23.39 | 28.30 | 10.48 | 11.05 | 24.79 | 36.41 | 35.11 | 19.31 | 11.89 | 37.08 | 21.47 |
| distilled student | 12.32 | 16.03 | 21.55 | 6.74 | 9.10 | 14.06 | 20.91 | 24.23 | 14.35 | 9.30 | 24.83 | 12.29 |
| improvement (%) | -1.2 | -0.1 | -0.1 | -1.9 | +2.4 | -3.8 | -0.6 | +1.0 | -0.6 | -0.6 | -0.2 | -1.3 |
| → *0.7B arch. 50B tokens teacher config ($\alpha$=0.2)* | | | | | | | | | | | | |
| pretrained teacher | 14.91 | 20.62 | 26.06 | 8.65 | 10.54 | 19.86 | 30.64 | 31.23 | 17.66 | 10.98 | 32.49 | 17.37 |
| distilled student | 12.21 | 15.98 | 21.43 | 6.53 | 9.01 | 13.02 | 19.37 | 24.13 | 14.30 | 9.23 | 24.49 | 12.18 |
| improvement (%) | -0.3 | +0.2 | +0.5 | +1.2 | +3.3 | +3.9 | +6.8 | +1.4 | -0.3 | +0.1 | +1.1 | -0.4 |
| → *0.7B arch. 80B tokens teacher config ($\alpha$=0.4)* | | | | | | | | | | | | |
| pretrained teacher | 14.01 | 19.35 | 24.54 | 8.02 | 10.40 | 15.52 | 23.99 | 29.33 | 16.49 | 10.40 | 31.18 | 12.83 |
| distilled student | 12.33 | 16.15 | 21.47 | 6.49 | 8.88 | 12.74 | 18.40 | 24.60 | 14.36 | 9.27 | 24.48 | 11.52 |
| improvement (%) | -1.3 | -0.9 | +0.3 | +1.8 | +4.7 | +5.9 | +11.5 | -0.5 | -0.7 | -0.3 | +1.2 | +5.0 |
| → *0.7B arch. 100B tokens teacher config ($\alpha$=0.2)* | | | | | | | | | | | | |
| pretrained teacher | 13.67 | 18.87 | 23.98 | 7.95 | 10.10 | 15.30 | 22.76 | 28.56 | 16.03 | 10.19 | 30.77 | 11.44 |
| distilled student | 12.13 | 15.74 | 21.26 | 6.53 | 8.87 | 12.79 | 19.18 | 24.27 | 14.17 | 9.15 | 24.10 | 11.31 |
| improvement (%) | +0.4 | +1.7 | +1.3 | +1.2 | +4.9 | +5.6 | +7.8 | +0.9 | +0.7 | +1.0 | +2.7 | +6.8 |
| → *0.7B arch. 300B tokens teacher config ($\alpha$=0.6)* | | | | | | | | | | | | |
| pretrained teacher | 12.49 | 17.05 | 22.08 | 7.17 | 10.44 | 14.15 | 19.84 | 25.70 | 14.71 | 9.43 | 28.50 | 8.39 |
| distilled student | 12.21 | 15.94 | 21.28 | 6.46 | 8.77 | 12.76 | 18.65 | 24.09 | 14.22 | 9.16 | 24.04 | 10.15 |
| improvement (%) | -0.3 | +0.5 | +1.2 | +2.4 | +5.9 | +5.8 | +10.3 | +1.6 | +0.3 | +0.9 | +2.9 | +16.3 |
| → *1.7B arch. 10B tokens teacher config ($\alpha$=0.2)* | | | | | | | | | | | | |
| pretrained teacher | 16.77 | 23.56 | 29.12 | 11.10 | 11.43 | 26.62 | 43.59 | 35.87 | 19.87 | 12.18 | 35.05 | 41.09 |
| distilled student | 12.34 | 16.12 | 21.62 | 6.83 | 9.18 | 13.58 | 20.46 | 24.65 | 14.44 | 9.32 | 24.45 | 13.23 |
| improvement (%) | -1.4 | -0.7 | -0.4 | -3.3 | +1.6 | -0.3 | +1.6 | -0.7 | -1.2 | -0.8 | +1.3 | -9.1 |
| → *1.7B arch. 30B tokens teacher config ($\alpha$=0.2)* | | | | | | | | | | | | |
| pretrained teacher | 13.14 | 17.76 | 23.13 | 7.39 | 9.85 | 16.29 | 23.70 | 26.95 | 15.47 | 9.88 | 27.04 | 14.69 |
| distilled student | 12.11 | 15.79 | 21.28 | 6.48 | 9.08 | 13.53 | 20.51 | 24.27 | 14.16 | 9.16 | 23.95 | 11.92 |
| improvement (%) | +0.5 | +1.4 | +1.2 | +2.1 | +2.6 | +0.1 | +1.4 | +0.9 | +0.7 | +0.9 | +3.3 | +1.7 |
| → *1.7B arch. 50B tokens teacher config ($\alpha$=0.4)* | | | | | | | | | | | | |
| pretrained teacher | 12.13 | 16.08 | 21.50 | 6.92 | 9.45 | 13.60 | 19.64 | 24.21 | 14.25 | 9.27 | 25.08 | 11.74 |
| distilled student | 12.05 | 15.66 | 21.08 | 6.41 | 9.07 | 13.24 | 18.96 | 23.93 | 14.06 | 9.09 | 23.78 | 11.16 |
| improvement (%) | +1.0 | +2.2 | +2.1 | +3.0 | +2.7 | +2.2 | +8.8 | +2.3 | +1.5 | +1.7 | +4.0 | +8.0 |
| → *1.7B arch. 80B tokens teacher config ($\alpha$=0.6)* | | | | | | | | | | | | |
| pretrained teacher | 11.46 | 15.01 | 20.42 | 6.17 | 9.31 | 12.31 | 18.19 | 22.96 | 13.48 | 8.82 | 23.53 | 8.83 |
| distilled student | 11.98 | 15.54 | 20.91 | 6.41 | 8.89 | 12.50 | 18.36 | 23.74 | 13.93 | 9.04 | 23.91 | 10.34 |
| improvement (%) | +1.6 | +2.9 | +2.9 | +3.0 | +4.7 | +7.7 | +11.7 | +3.0 | +2.3 | +2.2 | +3.5 | +14.8 |
| → *1.7B arch. 100B tokens teacher config ($\alpha$=0.6)* | | | | | | | | | | | | |
| pretrained teacher | 11.19 | 14.64 | 19.97 | 6.03 | 9.65 | 11.49 | 15.44 | 22.55 | 13.20 | 8.65 | 23.27 | 8.18 |
| distilled student | 11.93 | 15.60 | 20.83 | 6.29 | 9.22 | 12.22 | 17.74 | 23.54 | 13.87 | 8.98 | 23.85 | 10.07 |
| improvement (%) | +2.0 | +2.6 | +3.3 | +4.9 | +1.1 | +9.8 | +14.7 | +3.8 | +2.8 | +2.9 | +3.7 | +17.0 |
| → *1.7B arch. 300B tokens teacher config ($\alpha$=0.6)* | | | | | | | | | | | | |
| pretrained teacher | 10.17 | 13.17 | 18.50 | 5.40 | 10.38 | 10.17 | 14.82 | 20.48 | 12.16 | 8.07 | 21.46 | 6.11 |
| distilled student | 11.77 | 15.34 | 20.65 | 6.25 | 9.19 | 11.89 | 17.40 | 23.32 | 13.71 | 8.91 | 23.56 | 9.44 |
| improvement (%) | +3.3 | +4.2 | +4.1 | +5.5 | +1.5 | +12.2 | +16.3 | +4.8 | +3.9 | +3.7 | +4.9 | +22.2 |
| → *3.8B arch. 10B tokens teacher config ($\alpha$=0.2)* | | | | | | | | | | | | |
| pretrained teacher | 14.43 | 19.48 | 25.18 | 8.70 | 10.31 | 22.34 | 34.10 | 29.85 | 16.99 | 10.67 | 29.79 | 34.46 |
| distilled student | 12.22 | 15.91 | 21.43 | 6.57 | 8.91 | 12.94 | 20.24 | 24.45 | 14.30 | 9.23 | 23.96 | 12.60 |

*Continued on next page*

*Table 11 continued from previous page*

| perplexity (↓) | FW | Wiki | C4 | GSM | DMM | HEval | CSN | AX | CNN | ECHR | PM | XQA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| improvement (%) | -0.4 | +0.6 | +0.5 | +0.7 | +4.5 | +4.5 | +2.6 | +0.2 | -0.2 | +0.1 | +3.3 | -3.9 |
| → *3.8B arch. 30B tokens teacher config ($\alpha$=0.5)* | | | | | | | | | | | | |
| pretrained teacher | 11.20 | 14.53 | 20.04 | 6.19 | 8.63 | 12.32 | 18.08 | 22.28 | 13.25 | 8.71 | 22.98 | 11.27 |
| distilled student | 11.94 | 15.59 | 20.90 | 6.33 | 8.85 | 12.59 | 18.94 | 23.80 | 13.89 | 9.03 | 23.72 | 11.43 |
| improvement (%) | +1.9 | +2.6 | +2.9 | +4.2 | +5.1 | +7.1 | +8.9 | +2.8 | +2.6 | +2.3 | +4.2 | +5.8 |
| → *3.8B arch. 50B tokens teacher config ($\alpha$=0.5)* | | | | | | | | | | | | |
| pretrained teacher | 10.30 | 13.22 | 18.71 | 5.64 | 9.03 | 11.08 | 15.08 | 20.33 | 12.28 | 8.16 | 20.90 | 8.38 |
| distilled student | 11.83 | 15.44 | 20.78 | 6.40 | 9.14 | 12.02 | 18.01 | 23.66 | 13.77 | 8.97 | 23.51 | 10.24 |
| improvement (%) | +2.8 | +3.6 | +3.5 | +3.3 | +2.0 | +11.2 | +13.4 | +3.4 | +3.5 | +2.9 | +5.1 | +15.6 |
| → *3.8B arch. 80B tokens teacher config ($\alpha$=0.8)* | | | | | | | | | | | | |
| pretrained teacher | 9.64 | 12.18 | 17.77 | 5.23 | 8.65 | 9.10 | 12.72 | 18.89 | 11.60 | 7.81 | 20.07 | 6.92 |
| distilled student | 11.75 | 15.36 | 20.58 | 6.21 | 8.80 | 12.03 | 17.32 | 23.84 | 13.64 | 8.89 | 23.54 | 9.55 |
| improvement (%) | +3.5 | +4.0 | +4.4 | +6.0 | +5.6 | +11.2 | +16.7 | +2.6 | +4.4 | +3.9 | +4.9 | +21.2 |
| → *3.8B arch. 100B tokens teacher config ($\alpha$=0.8)* | | | | | | | | | | | | |
| pretrained teacher | 9.36 | 11.86 | 17.39 | 5.06 | 8.79 | 9.05 | 12.07 | 18.33 | 11.35 | 7.66 | 19.93 | 6.52 |
| distilled student | 11.70 | 15.27 | 20.55 | 6.42 | 9.05 | 12.07 | 17.17 | 23.25 | 13.62 | 8.88 | 23.37 | 9.61 |
| improvement (%) | +3.9 | +4.7 | +4.6 | +2.9 | +2.9 | +10.9 | +17.4 | +5.1 | +4.5 | +3.9 | +5.6 | +20.8 |
| → *3.8B arch. 300B tokens teacher config ($\alpha$=0.8)* | | | | | | | | | | | | |
| pretrained teacher | 8.29 | 10.61 | 16.12 | 4.71 | 8.83 | 8.28 | 11.56 | 16.88 | 10.53 | 7.17 | 20.29 | 5.16 |
| distilled student | 11.65 | 15.24 | 20.54 | 6.24 | 9.08 | 12.25 | 17.88 | 23.47 | 13.60 | 8.89 | 23.45 | 9.48 |
| improvement (%) | +4.3 | +4.8 | +4.6 | +5.6 | +2.7 | +9.6 | +14.0 | +4.2 | +4.7 | +3.9 | +5.3 | +21.9 |
| → *8.0B arch. 10B tokens teacher config ($\alpha$=0.4)* | | | | | | | | | | | | |
| pretrained teacher | 12.81 | 16.80 | 22.57 | 7.17 | 9.69 | 17.80 | 27.11 | 25.91 | 15.04 | 9.68 | 25.21 | 26.55 |
| distilled student | 12.23 | 15.95 | 21.33 | 6.50 | 8.97 | 13.01 | 18.43 | 24.20 | 14.23 | 9.19 | 23.88 | 13.40 |
| improvement (%) | -0.5 | +0.4 | +0.9 | +1.8 | +3.8 | +3.9 | +11.3 | +1.2 | +0.3 | +0.5 | +3.6 | -10.5 |
| → *8.0B arch. 30B tokens teacher config ($\alpha$=0.5)* | | | | | | | | | | | | |
| pretrained teacher | 9.98 | 12.83 | 18.31 | 5.53 | 8.60 | 10.21 | 15.07 | 19.79 | 11.98 | 8.04 | 20.32 | 9.57 |
| distilled student | 11.82 | 15.46 | 20.77 | 6.29 | 8.99 | 12.50 | 18.50 | 23.87 | 13.80 | 8.98 | 23.24 | 10.83 |
| improvement (%) | +2.9 | +3.5 | +3.5 | +4.8 | +3.6 | +7.7 | +11.0 | +2.5 | +3.3 | +2.8 | +6.2 | +10.7 |
| → *8.0B arch. 50B tokens teacher config ($\alpha$=0.8)* | | | | | | | | | | | | |
| pretrained teacher | 9.08 | 11.48 | 17.18 | 5.25 | 9.71 | 8.99 | 13.26 | 17.90 | 11.19 | 7.60 | 19.15 | 7.42 |
| distilled student | 11.70 | 15.36 | 20.61 | 6.36 | 9.18 | 11.89 | 17.42 | 23.52 | 13.62 | 8.90 | 23.32 | 9.74 |
| improvement (%) | +3.9 | +4.0 | +4.3 | +3.8 | +1.6 | +12.2 | +16.2 | +3.9 | +4.5 | +3.8 | +5.8 | +19.7 |
| → *8.0B arch. 80B tokens teacher config ($\alpha$=0.6)* | | | | | | | | | | | | |
| pretrained teacher | 8.38 | 10.67 | 16.38 | 4.82 | 9.74 | 8.24 | 11.66 | 16.98 | 10.64 | 7.28 | 18.28 | 6.00 |
| distilled student | 11.70 | 15.30 | 20.66 | 6.28 | 8.87 | 11.77 | 17.93 | 23.36 | 13.64 | 8.90 | 23.72 | 9.58 |
| improvement (%) | +3.9 | +4.4 | +4.0 | +5.1 | +4.8 | +13.1 | +13.8 | +4.6 | +4.4 | +3.7 | +4.2 | +21.0 |
| → *8.0B arch. 100B tokens teacher config ($\alpha$=0.8)* | | | | | | | | | | | | |
| pretrained teacher | 8.08 | 10.31 | 15.99 | 4.71 | 9.34 | 7.95 | 10.58 | 16.38 | 10.42 | 7.15 | 17.84 | 5.67 |
| distilled student | 11.66 | 15.27 | 20.60 | 6.24 | 8.97 | 11.28 | 16.29 | 23.32 | 13.58 | 8.89 | 23.72 | 9.25 |
| improvement (%) | +4.2 | +4.6 | +4.3 | +5.7 | +3.8 | +16.7 | +21.7 | +4.8 | +4.8 | +3.9 | +4.2 | +23.8 |
| → *8.0B arch. 300B tokens teacher config ($\alpha$=0.6)* | | | | | | | | | | | | |
| pretrained teacher | 6.77 | 8.85 | 14.89 | 4.36 | 9.43 | 8.21 | 11.17 | 15.25 | 9.74 | 6.73 | 17.42 | 4.62 |
| distilled student | 11.72 | 15.28 | 20.69 | 6.27 | 9.12 | 11.90 | 16.94 | 23.45 | 13.68 | 8.92 | 24.18 | 9.91 |
| improvement (%) | +3.8 | +4.5 | +3.9 | +5.1 | +2.2 | +12.2 | +18.5 | +4.2 | +4.1 | +3.5 | +2.3 | +18.3 |

Table 12: **Full downstream accuracy evaluation results.** We include MMLU (MMLU), ARC-Easy (ARC), SciQ (SQ), OpenBookQA (OBA), MathQA (MQA), TruthfulQA (TQA), ANLI-R1 (ANLI), CommonsenseQA (CSQA), HellaSwag (HS), PIQA (PIQA), WinoGrande (WG), Social IQa (SIQA), LogiQA 2.0 (LQ), MedMCQA (MCQA), and RACE (RACE).

| accuracy (↑) | MMLU | ARC | SQ | OBA | MQA | TQA | ANLI | CSQA | HS | PIQA | WG | SIQA | LQ | MCQA | RACE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| → *random chance* | | | | | | | | | | | | | | | |
| | 25.0 | 25.0 | 25.0 | 25.0 | 20.0 | – | 33.3 | 20.0 | 25.0 | 50.0 | 50.0 | 33.3 | 25.0 | 25.0 | 25.0 |
| → *standard pretrained student baseline* | | | | | | | | | | | | | | | |

*Continued on next page*

*Table 12 continued from previous page*

| accuracy (↑) | MMLU | ARC | SQ | OBA | MQA | TQA | ANLI | CSQA | HS | PIQA | WG | SIQA | LQ | MCQA | RACE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 25.4 | 62.8 | 87.2 | 33.4 | 23.7 | 21.4 | 32.9 | 18.9 | 51.1 | 71.1 | 54.0 | 38.7 | 23.6 | 24.5 | 32.7 |
| *→ 0.7B arch. 10B tokens teacher config (α=0.2)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.8 | 47.3 | 73.1 | 29.6 | 21.5 | 17.9 | 33.3 | 19.2 | 32.4 | 62.5 | 50.7 | 35.7 | 22.6 | 23.0 | 28.4 |
| distilled student | 25.8 | 60.3 | 86.1 | 32.2 | 22.7 | 20.6 | 33.7 | 19.5 | 47.5 | 70.6 | 53.7 | 40.1 | 22.2 | 27.8 | 32.4 |
| improvement (%) | +1.7 | -4.0 | -1.3 | -3.6 | -4.1 | -4.0 | +2.4 | +3.0 | -7.2 | -0.7 | -0.4 | +3.6 | -5.9 | +13.2 | -0.9 |
| *→ 0.7B arch. 30B tokens teacher config (α=0.2)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.6 | 53.0 | 80.5 | 31.2 | 22.1 | 21.8 | 33.0 | 20.6 | 39.2 | 66.2 | 52.6 | 37.6 | 24.1 | 27.4 | 30.3 |
| distilled student | 26.5 | 60.7 | 87.1 | 34.2 | 23.6 | 21.9 | 34.9 | 20.6 | 49.2 | 70.9 | 54.2 | 40.3 | 22.8 | 24.8 | 32.9 |
| improvement (%) | +4.2 | -3.4 | -0.1 | +2.4 | -0.1 | +2.3 | +6.1 | +9.1 | -3.8 | -0.2 | +0.4 | +4.1 | -3.2 | +1.1 | +0.6 |
| *→ 0.7B arch. 50B tokens teacher config (α=0.2)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.1 | 55.6 | 81.6 | 33.0 | 22.4 | 18.4 | 33.2 | 20.3 | 43.0 | 67.8 | 52.1 | 38.6 | 23.5 | 24.5 | 31.3 |
| distilled student | 25.7 | 60.7 | 86.5 | 32.6 | 23.1 | 19.1 | 34.4 | 20.2 | 49.9 | 71.2 | 55.6 | 40.8 | 23.2 | 27.7 | 32.7 |
| improvement (%) | +1.1 | -3.3 | -0.8 | -2.4 | -2.3 | -10.9 | +4.6 | +6.9 | -2.3 | +0.2 | +2.9 | +5.4 | -1.6 | +12.8 | 0.0 |
| *→ 0.7B arch. 80B tokens teacher config (α=0.4)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.5 | 58.8 | 84.4 | 29.8 | 22.7 | 18.1 | 33.9 | 19.2 | 45.5 | 70.1 | 51.1 | 37.0 | 23.7 | 25.3 | 31.8 |
| distilled student | 25.8 | 60.6 | 85.9 | 33.4 | 22.7 | 20.3 | 33.8 | 19.7 | 49.2 | 69.9 | 55.2 | 40.3 | 23.8 | 27.9 | 33.9 |
| improvement (%) | +1.4 | -3.5 | -1.5 | 0.0 | -3.8 | -5.1 | +2.7 | +3.9 | -3.8 | -1.7 | +2.3 | +4.2 | +0.8 | +13.6 | +3.5 |
| *→ 0.7B arch. 100B tokens teacher config (α=0.4)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.0 | 59.6 | 83.3 | 30.6 | 23.5 | 17.9 | 32.8 | 19.3 | 46.7 | 69.5 | 53.1 | 37.5 | 24.2 | 26.5 | 32.2 |
| distilled student | 27.1 | 61.2 | 85.9 | 32.4 | 23.8 | 21.8 | 33.7 | 19.8 | 49.6 | 70.7 | 53.7 | 40.2 | 24.2 | 24.0 | 32.8 |
| improvement (%) | +6.7 | -2.5 | -1.5 | -3.0 | +0.4 | +1.7 | +2.4 | +4.8 | -2.9 | -0.5 | -0.6 | +3.8 | +2.7 | -1.9 | +0.3 |
| *→ 0.7B arch. 300B tokens teacher config (α=0.6)* | | | | | | | | | | | | | | | |
| pretrained teacher | 24.9 | 62.7 | 86.7 | 33.2 | 25.3 | 17.0 | 33.5 | 17.7 | 50.9 | 70.6 | 55.2 | 39.7 | 23.3 | 28.0 | 33.1 |
| distilled student | 26.3 | 61.1 | 85.5 | 34.4 | 23.7 | 21.8 | 33.9 | 20.1 | 50.1 | 70.9 | 57.1 | 40.2 | 24.7 | 23.4 | 33.2 |
| improvement (%) | +3.6 | -2.7 | -1.9 | +3.0 | +0.1 | +1.7 | +3.0 | +6.1 | -2.0 | -0.2 | +5.7 | +3.8 | +4.9 | -4.5 | +1.5 |
| *→ 1.7B arch. 10B tokens teacher config (α=0.2)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.0 | 50.5 | 79.5 | 29.4 | 22.2 | 22.3 | 33.8 | 20.6 | 38.0 | 65.5 | 51.5 | 36.9 | 22.4 | 26.5 | 30.3 |
| distilled student | 26.6 | 59.8 | 85.0 | 31.8 | 23.9 | 21.4 | 31.3 | 19.2 | 48.8 | 70.9 | 52.7 | 39.9 | 23.0 | 27.0 | 32.3 |
| improvement (%) | +4.7 | -4.8 | -2.5 | -4.8 | +1.1 | 0.0 | -4.9 | +1.7 | -4.4 | -0.2 | -2.3 | +3.2 | -2.4 | +10.1 | -1.2 |
| *→ 1.7B arch. 30B tokens teacher config (α=0.2)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.0 | 60.0 | 85.5 | 34.2 | 23.8 | 20.6 | 32.7 | 20.1 | 47.4 | 70.0 | 52.8 | 39.8 | 23.4 | 23.7 | 32.0 |
| distilled student | 26.3 | 61.7 | 85.2 | 33.2 | 23.3 | 19.5 | 32.9 | 22.0 | 50.4 | 71.2 | 54.5 | 40.3 | 23.7 | 27.1 | 33.5 |
| improvement (%) | +3.8 | -1.8 | -2.3 | -0.6 | -1.4 | -9.1 | 0.0 | +16.5 | -1.4 | +0.2 | +0.9 | +4.1 | +0.3 | +10.3 | +2.3 |
| *→ 1.7B arch. 50B tokens teacher config (α=0.2)* | | | | | | | | | | | | | | | |
| pretrained teacher | 27.0 | 60.2 | 86.8 | 33.6 | 22.0 | 19.3 | 33.9 | 19.7 | 51.2 | 71.3 | 55.0 | 40.8 | 22.2 | 22.3 | 32.7 |
| distilled student | 26.8 | 62.3 | 86.9 | 34.8 | 23.2 | 19.7 | 34.6 | 19.3 | 51.1 | 70.9 | 55.5 | 41.1 | 24.7 | 25.4 | 32.3 |
| improvement (%) | +5.4 | -0.8 | -0.3 | +4.2 | -1.7 | -8.0 | +5.2 | +2.2 | +0.0 | -0.2 | +2.8 | +6.2 | +4.9 | +3.4 | -1.2 |
| *→ 1.7B arch. 80B tokens teacher config (α=0.2)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.4 | 64.1 | 89.5 | 37.2 | 24.5 | 23.5 | 34.5 | 20.5 | 54.2 | 72.4 | 54.9 | 41.1 | 24.4 | 27.4 | 34.7 |
| distilled student | 26.3 | 63.3 | 87.5 | 35.8 | 24.1 | 22.5 | 34.8 | 19.2 | 51.5 | 71.9 | 55.7 | 41.4 | 22.1 | 22.7 | 33.4 |
| improvement (%) | +3.5 | +0.8 | +0.3 | +7.2 | +2.0 | +5.1 | +5.8 | +1.3 | +0.7 | +1.1 | +3.2 | +7.0 | -6.2 | -7.4 | +2.0 |
| *→ 1.7B arch. 100B tokens teacher config (α=0.5)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.7 | 66.5 | 89.7 | 37.6 | 22.7 | 22.8 | 35.9 | 21.2 | 55.7 | 72.0 | 56.5 | 39.9 | 23.8 | 24.0 | 34.5 |
| distilled student | 25.5 | 62.5 | 87.7 | 36.4 | 23.4 | 19.0 | 33.6 | 20.5 | 51.6 | 72.4 | 54.9 | 40.5 | 23.7 | 26.8 | 33.8 |
| improvement (%) | +0.3 | -0.4 | +0.6 | +9.0 | -1.1 | -11.4 | +2.1 | +8.2 | +0.9 | +1.8 | +1.8 | +4.6 | +0.3 | +9.2 | +3.2 |
| *→ 1.7B arch. 300B tokens teacher config (α=0.8)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.1 | 67.5 | 93.1 | 40.0 | 23.8 | 20.7 | 35.0 | 21.9 | 59.4 | 73.3 | 58.7 | 41.0 | 26.3 | 26.8 | 34.8 |
| distilled student | 26.6 | 64.7 | 88.3 | 36.8 | 25.0 | 21.8 | 35.9 | 19.2 | 52.6 | 71.7 | 57.8 | 39.6 | 24.2 | 26.3 | 34.3 |
| improvement (%) | +4.8 | +3.1 | +1.3 | +10.2 | +5.7 | +1.7 | +9.1 | +1.3 | +2.9 | +0.8 | +7.0 | +2.2 | +2.7 | +7.4 | +4.7 |
| *→ 3.8B arch. 10B tokens teacher config (α=0.5)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.6 | 57.1 | 82.5 | 32.4 | 22.7 | 22.9 | 33.1 | 19.7 | 43.4 | 68.0 | 51.9 | 38.3 | 22.3 | 25.4 | 32.5 |
| distilled student | 26.5 | 59.5 | 85.9 | 31.2 | 23.5 | 23.1 | 34.4 | 18.7 | 47.1 | 70.7 | 52.3 | 39.7 | 23.0 | 24.2 | 32.5 |
| improvement (%) | +4.2 | -5.2 | -1.5 | -6.6 | -0.8 | +8.0 | +4.6 | -1.3 | -7.9 | -0.5 | -3.1 | +2.5 | -2.7 | -1.4 | -0.6 |
| *→ 3.8B arch. 30B tokens teacher config (α=0.5)* | | | | | | | | | | | | | | | |
| pretrained teacher | 24.8 | 66.8 | 89.4 | 38.6 | 22.5 | 18.5 | 35.5 | 19.9 | 54.8 | 72.3 | 55.7 | 40.8 | 22.7 | 24.3 | 35.2 |
| distilled student | 26.7 | 63.0 | 88.0 | 35.4 | 22.7 | 21.9 | 34.0 | 19.9 | 51.5 | 71.8 | 55.7 | 41.6 | 23.5 | 26.8 | 32.1 |
| improvement (%) | +5.2 | +0.3 | +0.9 | +6.0 | -4.0 | +2.3 | +3.3 | +5.2 | +0.7 | +1.1 | +3.2 | +7.5 | -0.3 | +9.5 | -2.0 |
| *→ 3.8B arch. 50B tokens teacher config (α=0.6)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.3 | 67.3 | 89.4 | 39.4 | 25.0 | 19.7 | 35.1 | 19.0 | 58.9 | 74.2 | 58.2 | 42.4 | 23.5 | 30.2 | 35.4 |

*Continued on next page*

*Table 12 continued from previous page*

| accuracy (↑) | MMLU | ARC | SQ | OBA | MQA | TQA | ANLI | CSQA | HS | PIQA | WG | SIQA | LQ | MCQA | RACE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| distilled student | 25.9 | 65.5 | 88.9 | 35.6 | 23.7 | 22.4 | 34.9 | 18.9 | 52.6 | 72.4 | 55.9 | 41.2 | 23.5 | 26.8 | 33.6 |
| improvement (%) | +1.9 | +4.4 | +1.9 | +6.6 | +0.1 | +4.6 | +6.1 | 0.0 | +3.0 | +1.9 | +3.5 | +6.5 | -0.5 | +9.2 | +2.6 |
| → *3.8B arch. 80B tokens teacher config (α=0.8)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.3 | 70.3 | 92.1 | 41.4 | 25.4 | 21.8 | 33.2 | 20.1 | 61.9 | 74.6 | 59.9 | 42.0 | 23.9 | 26.3 | 35.8 |
| distilled student | 25.8 | 64.6 | 88.5 | 35.2 | 24.2 | 20.3 | 34.0 | 21.5 | 52.9 | 72.3 | 56.3 | 40.1 | 23.9 | 27.4 | 33.6 |
| improvement (%) | +1.5 | +2.8 | +1.5 | +5.4 | +2.3 | -5.1 | +3.3 | +13.4 | +3.5 | +1.7 | +4.2 | +3.6 | +1.1 | +11.9 | +2.6 |
| → *3.8B arch. 100B tokens teacher config (α=0.6)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.6 | 70.3 | 92.4 | 40.0 | 25.2 | 21.3 | 33.9 | 17.9 | 63.2 | 75.2 | 62.4 | 42.7 | 23.2 | 26.4 | 36.4 |
| distilled student | 25.1 | 65.1 | 88.8 | 37.4 | 23.7 | 22.6 | 32.0 | 19.7 | 52.9 | 72.1 | 57.5 | 41.2 | 23.7 | 25.7 | 33.8 |
| improvement (%) | -1.3 | +3.6 | +1.8 | +12.0 | +0.3 | +5.7 | -2.7 | +3.9 | +3.5 | +1.5 | +6.6 | +6.5 | +0.3 | +5.0 | +3.2 |
| → *3.8B arch. 300B tokens teacher config (α=0.5)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.6 | 72.1 | 94.1 | 42.8 | 26.9 | 25.3 | 34.0 | 19.5 | 66.8 | 76.9 | 66.1 | 42.8 | 23.0 | 29.0 | 36.4 |
| distilled student | 25.1 | 65.7 | 88.6 | 37.4 | 23.6 | 21.4 | 32.6 | 19.4 | 53.1 | 71.9 | 56.6 | 41.2 | 23.5 | 28.3 | 33.5 |
| improvement (%) | -1.0 | +4.6 | +1.6 | +12.0 | -0.4 | 0.0 | -0.9 | +2.6 | +3.8 | +1.2 | +4.8 | +6.5 | -0.5 | +15.5 | +2.3 |
| → *8.0B arch. 10B tokens teacher config (α=0.4)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.2 | 59.6 | 85.5 | 33.6 | 23.7 | 20.1 | 31.9 | 20.1 | 48.4 | 70.5 | 53.1 | 38.7 | 22.3 | 25.8 | 32.3 |
| distilled student | 26.7 | 60.9 | 85.9 | 34.4 | 23.4 | 22.5 | 33.4 | 18.4 | 49.8 | 70.8 | 53.4 | 40.8 | 22.3 | 26.7 | 33.3 |
| improvement (%) | +5.1 | -3.1 | -1.5 | +3.0 | -1.1 | +5.1 | +1.5 | -2.6 | -2.5 | -0.4 | -1.2 | +5.4 | -5.4 | +9.0 | +1.8 |
| → *8.0B arch. 30B tokens teacher config (α=0.6)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.4 | 66.8 | 90.5 | 38.2 | 24.5 | 22.4 | 35.1 | 19.3 | 59.6 | 75.0 | 59.7 | 40.7 | 24.2 | 25.1 | 34.7 |
| distilled student | 26.0 | 64.1 | 87.1 | 33.2 | 24.5 | 23.6 | 34.4 | 22.3 | 52.7 | 72.0 | 56.0 | 40.7 | 24.0 | 26.4 | 33.4 |
| improvement (%) | +2.2 | +2.1 | -0.1 | -0.6 | +3.4 | +10.3 | +4.6 | +17.7 | +3.2 | +1.3 | +3.8 | +5.2 | +1.9 | +7.7 | +2.0 |
| → *8.0B arch. 50B tokens teacher config (α=0.8)* | | | | | | | | | | | | | | | |
| pretrained teacher | 25.1 | 68.8 | 91.6 | 40.8 | 24.4 | 22.2 | 34.9 | 19.7 | 63.1 | 75.9 | 59.4 | 42.3 | 23.8 | 26.7 | 37.1 |
| distilled student | 26.6 | 63.2 | 87.1 | 34.8 | 23.2 | 20.8 | 34.0 | 20.3 | 53.2 | 72.4 | 57.2 | 42.6 | 24.0 | 26.0 | 33.5 |
| improvement (%) | +4.8 | +0.6 | -0.1 | +4.2 | -1.7 | -2.9 | +3.3 | +7.4 | +4.0 | +1.9 | +6.0 | +10.1 | +1.6 | +6.0 | +2.3 |
| → *8.0B arch. 80B tokens teacher config (α=1.0)* | | | | | | | | | | | | | | | |
| pretrained teacher | 24.8 | 72.7 | 93.1 | 42.4 | 25.8 | 22.6 | 33.3 | 19.8 | 65.7 | 77.1 | 62.7 | 42.4 | 25.3 | 30.9 | 34.7 |
| distilled student | 26.2 | 64.2 | 88.6 | 37.2 | 23.9 | 20.4 | 34.4 | 20.9 | 53.2 | 71.9 | 54.6 | 40.9 | 23.6 | 28.0 | 33.7 |
| improvement (%) | +3.3 | +2.2 | +1.6 | +11.4 | +1.0 | -4.6 | +4.6 | +10.4 | +4.1 | +1.1 | +1.2 | +5.7 | 0.0 | +14.3 | +2.9 |
| → *8.0B arch. 100B tokens teacher config (α=1.0)* | | | | | | | | | | | | | | | |
| pretrained teacher | 26.8 | 73.5 | 94.4 | 42.6 | 25.9 | 22.2 | 33.7 | 20.6 | 67.0 | 76.6 | 64.9 | 43.5 | 23.9 | 26.6 | 35.9 |
| distilled student | 26.3 | 62.2 | 88.3 | 37.2 | 24.8 | 22.2 | 34.3 | 18.5 | 53.7 | 72.0 | 55.2 | 41.4 | 23.0 | 26.2 | 33.8 |
| improvement (%) | +3.5 | -0.9 | +1.3 | +11.4 | +4.8 | +3.4 | +4.3 | -2.2 | +5.2 | +1.4 | +2.3 | +6.9 | -2.7 | +6.6 | +3.2 |
| → *8.0B arch. 300B tokens teacher config (α=0.8)* | | | | | | | | | | | | | | | |
| pretrained teacher | 27.2 | 76.5 | 95.1 | 43.8 | 27.5 | 27.3 | 33.9 | 20.8 | 70.6 | 77.4 | 65.4 | 43.8 | 24.2 | 28.1 | 37.9 |
| distilled student | 26.4 | 63.3 | 86.8 | 36.6 | 23.2 | 23.0 | 30.3 | 20.1 | 53.6 | 71.5 | 56.4 | 40.6 | 24.4 | 25.1 | 33.9 |
| improvement (%) | +3.8 | +0.9 | -0.5 | +9.6 | -1.8 | +7.4 | -7.9 | +6.1 | +4.9 | +0.6 | +4.4 | +5.0 | +3.5 | +2.2 | +3.5 |

## 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: The abstract and Section 1 state three empirical findings: (1) weak-to-strong and same-level distillation can improve over standard pretraining with proper loss mixing, (2) stronger teachers do not always produce better students, and (3) distillation improves generalization more readily than in-domain fit. Each finding is directly supported by the experiments in Sections 4, 5, and 6, with full per-benchmark results and additional analyses in the appendix.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Limitations and scope are discussed throughout the paper. The study focuses on logit-level distillation in LLM pretraining; Section 7 contrasts this with the teacher-generated synthetic data approach used by recent industrial models. Section 6 reports configurations where weak-to-strong distillation does not improve in-domain perplexity (especially for larger students). The grid search over the loss mixing coefficient $\alpha$ is identified in Section 5 as a practical cost. Robustness across model families is partially addressed via the Qwen3 ablation (Appendix G.1) and across data scale via the 300B-token ablation (Appendix G.2); the main results use a single corpus (FineWeb-Edu) and one student size (1.7B) as the primary point of study.

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: The paper is empirical and does not present formal theorems or proofs. The mathematical formulations of the language modeling, distillation, and mixed loss objectives in Section 2 are standard and stated as definitions, not theoretical results.

   Guidelines:

   - The answer [N/A] 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: Section 3 specifies the four architecture sizes, six teacher token budgets, the 50B-token student budget, optimizer settings (AdamW with $\beta_1$=0.9, $\beta_2$=0.95, weight decay 0.1, peak LR $3\times10^{-4}$ with cosine decay and 5% linear warmup), context length (8192), batch size (256), bf16 precision, and the distillation objective with the $\alpha$ grid. Appendix B provides full architecture configurations and Appendix C provides per-benchmark evaluation protocols. All datasets used (FineWeb-Edu and the 11 OOD corpora and 15 downstream benchmarks listed in Section 3.3) are publicly available.

   Guidelines:

   - The answer [N/A] 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: To preserve anonymity, code is not released with the submission. We plan to release training, distillation, and evaluation code upon acceptance. All datasets used (FineWeb-Edu, 11 out-of-distribution evaluation corpora, and 15 downstream benchmarks) are publicly available and cited in Section 3.3 and Appendix C, which together with Section 3 and Appendix B provide sufficient detail to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3.2 describes all training hyperparameters (optimizer, learning rate, schedule, warmup, batch size, context length, precision, data ordering). The $\alpha$ grid is reported in Section 3.2, and the selection methodology is explained in Sections 4 and 5, with a robustness check using a single jointly optimized $\alpha$ in Appendix E. Evaluation splits and protocols are detailed in Section 3.3 and Appendix C.

Guidelines:

- The answer [N/A] 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: Multi-seed baseline variance (mean and standard deviation across multiple training seeds for all four architecture sizes) is reported in Appendix H, showing that reported distillation improvements (typically 3–9%) exceed the seed-noise floor by an order of magnitude. Statistical significance for the hard-token concentration finding is reported via bootstrap with 1000 resamples in Appendix J.2 ($p < 0.001$ for all students with positive overall improvement).

   Guidelines:

   - The answer [N/A] 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: All experiments were run on a mix of TPU and GPU clusters. Section 3 fully specifies the experimental scope: four teacher architectures (0.7B–8.0B), six teacher token budgets (10B–300B), six $\alpha$ values per teacher–student pair, fixed 50B-token student training, and the additional ablations across student sizes (Section 6) and architecture families (Appendix G.1). Together with the architecture details in Appendix B, total compute can be estimated using standard $6N$ training-FLOPs accounting.

   Guidelines:

   - The answer [N/A] 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 reviewed the NeurIPS Code of Ethics. The work uses publicly available datasets and trains language models from scratch for the purpose of methodological study; no human subjects, deceptive practices, or restricted data are involved.

   Guidelines:

   - The answer [N/A] 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: The findings reduce the compute barrier to distillation pretraining by showing that smaller, less-trained teachers can suffice with proper loss mixing, which benefits academic and small-team practitioners and lowers energy use for a common training step. As foundational research on language model training, the work does not introduce new model capabilities beyond what already-public teacher and student architectures provide; we are not aware of direct paths to misuse beyond those that exist for general-purpose LLM research.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new pretrained models or scraped datasets. Trained models are research artifacts derived from publicly available data and are used solely for the analyses reported in the paper.

    Guidelines:

    - The answer [N/A] 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: All datasets and benchmarks used (FineWeb-Edu and the 11 OOD evaluation corpora and 15 downstream benchmarks) are publicly released and cited in Section 3.3 and Appendix C. The Llama3 architecture family and Qwen3 architecture family are described and used as model templates; teachers and students are trained from scratch on FineWeb-Edu and used only for analysis, with no redistribution of pretrained weights.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not release new datasets or model checkpoints with the submission.

    Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve human subjects research; IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the subject of study (we train and evaluate them), but they are not used as a non-standard component of the methodology. No LLM-generated content is used in the core analyses or results beyond the trained models themselves.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.