BEYOND PASSIVE IMITATION: ON-POLICY MID-TRAINING DISTILLATION FOR LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual pre-training (CPT) adapts large language models (LLMs) to specialized domains, yet its standard next-token prediction (NTP) objective treats every corpus token as a fixed target, regardless of its utility to the current model. In a controlled study on the same 1B-token mathematical corpus, we observe sharply different outcomes across base models: NTP improves Llama-3.2-3B's performance on mathematical benchmarks but substantially degrades Qwen3-1.7B-Base's performance, indicating that, under a fixed training recipe, the same training sequences can have model-dependent effects. To address this problem, we introduce **On-Policy Mid-Training Distillation (OPMD)**, which reformulates mid-training from offline text imitation into active, query-driven knowledge acquisition. At selected anchor positions, OPMD samples trajectories from the evolving student policy and applies dense token-level reverse-KL rewards from a frozen teacher, training the model on states it actually visits rather than only on ground-truth prefixes. This on-policy objective reduces the training–generation mismatch of teacher-forced NTP. On MATH-500 and Minerva, OPMD improves the Qwen student over its base performance and outperforms the evaluated NTP baseline, which degrades on both benchmarks. OPMD integrates into CPT pipelines without added inference-time overhead and offers a model-aware alternative to passive domain adaptation.

1 INTRODUCTION

Continual pre-training (CPT), often referred to as mid-training in recent large language model (LLM) pipelines, has become an effective paradigm for adapting general-purpose LLMs (Grattafiori et al., 2024; Yang et al., 2025; Xu et al., 2026) to specialized domains such as mathematics and code (Gururangan et al., 2020; Ke et al., 2023). By continuing autoregressive pre-training on domain-specific corpora, models can acquire substantial domain knowledge while preserving much of their broader capability (Shao et al., 2024). In practice, CPT typically continues to optimize the standard next-token prediction (NTP) objective over pre-collected text drawn from the target domain of interest. Under this teacher-forced objective, the model acts as an offline language imitator: it minimizes token-level cross-entropy against the next tokens in fixed, ground-truth sequences during training.

However, standard NTP-based mid-training follows a fundamentally passive learning dynamic. The model is optimized against a fixed, offline corpus without explicitly assessing how informative or useful each training sequence is for its current policy (Ying et al., 2025). Consequently, both the quality and composition of the corpus can substantially affect the resulting model. Recent studies of continual pre-training show that corpus quality, distribution, mixture, and redundancy influence its learning dynamics and downstream performance (Guo et al., 2025b; Chen et al., 2025). These observations expose a central limitation of NTP-based CPT: its effectiveness remains closely tied to which sequences enter the training corpus and how frequently they are sampled.

To examine this sensitivity empirically, we conduct a controlled pilot study with two base models that differ in their initial levels of mathematical capability: Llama-3.2-3B (Grattafiori et al., 2024) and Qwen3-1.7B-Base (Yang et al., 2025). We apply the same NTP mid-training recipe to both models using a 1B-token subset randomly sampled via streaming from MegaMath-Web-Pro. After that, we evaluate them on MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) and Minerva (Lewkowycz et al., 2022). Figure 1 reveals opposite outcomes under the same corpus and training objective: Llama-3.2-3B improves on both benchmarks, whereas Qwen3-1.7B-Base suffers

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

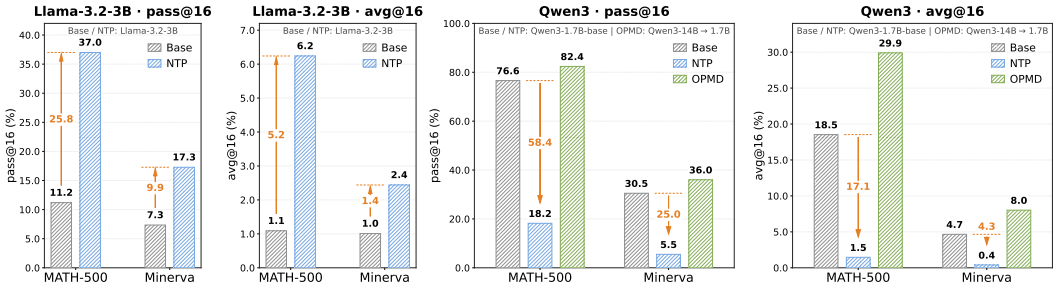


Figure 1: **Empirical validation of NTP mid-training data sensitivity across base models with different capabilities.** Under identical NTP CPT on a 1B-token subset streamed from MegaMath-Web-Pro, Llama-3.2-3B achieves accuracy gains, whereas Qwen3-1.7B-Base suffers substantial performance degradation on the MATH-500 and Minerva benchmarks.

substantial degradation on both. This contrast indicates that the utility of a fixed CPT corpus can depend strongly on the capability and pretraining state of each model.

We hypothesize that a stronger base model has more developed reasoning behaviors to preserve and may therefore be more vulnerable to conflicting or low-utility training signals. Current practice mainly addresses corpus sensitivity through static data curation, including heuristic filtering and mixture reweighting (Xie et al., 2023a; Chen et al., 2025). These methods can improve the training distribution, but they operate before optimization and do not respond to the states that the student model actually visits as training progresses. Teacher-forced NTP introduces a separate limitation: it trains on ground-truth prefixes rather than contexts generated by the model's own policy. This training–generation mismatch produces exposure bias (Ross et al., 2011; Gudibande et al., 2024).

Addressing these limitations requires a mid-training objective that can focus learning on informative contexts and supervise the model on its own trajectories. On-Policy Distillation (OPD) offers such a mechanism by sampling trajectory rollouts from the student's evolving policy π_θ and minimizing reverse KL divergence to a teacher policy π_ψ (Gu et al., 2024; Agarwal et al., 2024; Li et al., 2026). Because the student receives supervision on states induced by its own decisions during trajectory generation, OPD directly reduces the mismatch between training and generation. Existing OPD methods, however, primarily target post-training settings such as instruction-following alignment and reasoning capability enhancement. Their use during continual pre-training, especially for acquiring domain knowledge without eroding a capable base model, remains underexplored.

To address this gap, we introduce **On-Policy Mid-Training Distillation (OPMD)**, which reformulates mid-training as an active, query-driven knowledge acquisition process. OPMD first identifies high-uncertainty anchor positions along raw domain sequences and converts each preceding context into a rollout prompt. The student then generates a trajectory under its current policy π_θ, while a frozen teacher π_ψ provides dense token-level reverse-KL supervision on the states visited by that trajectory. This design moves beyond uniform imitation of corpus tokens by concentrating training on informative contexts and aligning the student where its own policy takes it. OPMD changes only the training procedure, so it can be integrated into existing CPT pipelines without a deployment-time teacher, extra model modules, or additional inference cost at deployment.

We make three contributions to continual pre-training through on-policy distillation:

1. **Controlled empirical finding.** We identify a model-dependent failure mode of passive CPT through a controlled pilot study. With the same 1B-token corpus and NTP training recipe, Llama-3.2-3B improves while Qwen3-1.7B-Base degrades on MATH-500 and Minerva. This contrast establishes that a fixed CPT corpus can have model-dependent utility.
2. **On-Policy Mid-Training Distillation.** We formulate `OPMD` for the continual pre-training stage by combining uncertainty-based anchor selection, rollouts from the evolving student policy, and dense token-level reverse-KL supervision from a frozen teacher. The resulting objective concentrates knowledge transfer on informative contexts and trains the student on states induced by its own generations rather than only on ground-truth corpus prefixes.

3. **Evaluation and analysis.** We evaluate OPMD on mathematical benchmarks against an NTP-based CPT baseline and across controlled anchor-selection variants with the same training-sample budget in each arm. The results show that OPMD improves the Qwen student over its base performance while avoiding the degradation observed with NTP mid-training. The anchor-selection comparisons examine how different rollout starting positions relate to these gains.

2 Related Work

Continual Pre-Training and Data Curation. Continual pre-training (CPT), or mid-training, serves as a standard paradigm for infusing domain-specific knowledge—such as mathematics and programming code—into general-purpose base language models (Gururangan et al., 2020; Ke et al., 2023). In the current literature and industrial practices, the vast majority of CPT approaches strictly adhere to the standard Next Token Prediction (NTP) autoregressive objective over target domain corpora (Lewkowycz et al., 2022; Shao et al., 2024; Grattafiori et al., 2024). To maximize model performance under the NTP objective, existing research heavily relies on data-centric curation paradigms to enhance corpus quality, which can be broadly categorized into: **(i)** *Heuristic and deduplication filtering*, which prunes low-quality, noisy, or redundant text using static rules or perplexity thresholds (Penedo et al., 2024); and **(ii)** *Distribution matching and domain reweighting*, which selects target-relevant domain subsets or optimizes domain mixture ratios prior to training (Xie et al., 2023b;a). Although these data curation methods effectively enhance corpus purity, they treat data processing as a static, passive pre-processing step independent of the model's active learning dynamics and the contexts it visits. Consequently, they fail to alter the model's rigid reliance on data quality at the level of the training mechanism during mid-training.

On-Policy Distillation in LLMs. Knowledge Distillation (KD) transfers parametric capabilities and reasoning patterns from massive teacher models to smaller student models (Kim & Rush, 2016; Gou et al., 2021; Gu et al., 2024). However, traditional KD approaches in LLMs predominantly operate in an off-policy manner on static datasets or teacher-generated traces (Sanh et al., 2020; Wang et al., 2025). As a result, the student is trained only in contexts frequented by the teacher rather than in states encountered under its own policy, introducing severe compounding errors during sequence generation: an early mistake made by the student places it in unfamiliar states unobserved during training, causing its policy to diverge rapidly from target behavior (Ross et al., 2011; Gudibande et al., 2024). To overcome this discrepancy, On-Policy Distillation (OPD)—exemplified by MiniLLM and Generalized Knowledge Distillation (GKD)—enables student models to sample trajectory rollouts under their own evolving policy π_θ while optimizing reverse KL divergence against a frozen teacher π_ψ, effectively teaching the student to recover from its own generation mistakes (Gu et al., 2024; Agarwal et al., 2024). Recent empirical investigations further analyze OPD's underlying mechanics, demonstrating how dense token-level reward assignment effectively stabilizes state exploration and mitigates error accumulation (Li et al., 2026). Overall, existing OPD methodologies mainly focus on either instruction-following alignment or model compression in task-specific post-training phases.

Unlike prior CPT strategies that passively adapt corpora to the rigid NTP loss, we propose On-Policy Mid-Training Distillation (OPMD). By introducing on-policy trajectory sampling and optimizing student exploration via token-level reverse KL rewards during continual pre-training, OPMD transforms mid-training into an active, query-driven knowledge acquisition process.

3 Methodology

3.1 Preliminaries

In this section, we formally define the problem setup for Continual Pre-Training (CPT) and review the mathematical formulations of On-Policy Distillation (OPD).

Continual Pre-Training via Next Token Prediction. Let $\mathcal{D}$ denote a target-domain pre-training corpus consisting of token sequences $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, where each token $x_t \in \mathcal{V}$ belongs to a vocabulary $\mathcal{V}$. Standard continual pre-training (CPT) optimizes a base language model parameterized by θ, denoted as policy $\pi_\theta(x_t \mid \mathbf{x}_{<t})$, using the causal Next Token Prediction (NTP) objective. The

standard autoregressive cross-entropy loss over corpus $\mathcal{D}$ is formulated as:

$$\mathcal{L}_{\text{NTP}}(\theta) = -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\sum_{t=1}^{T}\log \pi_\theta(x_t \mid \mathbf{x}_{<t})\right]. \tag{1}$$

While $\mathcal{L}_{\text{NTP}}$ forces the model to strictly imitate the empirical text distribution of $\mathcal{D}$, it treats all tokens with uniform loss weights regardless of token difficulty or data noise, making optimization highly sensitive to corpus quality.

On-Policy Distillation in LLMs. To overcome distribution shift inherent to off-policy sequence imitation, On-Policy Distillation (OPD) leverages a frozen teacher model π_ψ to guide a student model π_θ. Instead of calculating distillation targets over fixed ground-truth corpora, OPD allows the student model to generate trajectory rollouts sampled directly from its own evolving policy. Formally, using the reverse Kullback-Leibler (KL) divergence to mitigate state-drift and prevent over-generalization, the standard OPD objective is written as:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\,\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}\left[\sum_{k=1}^{K}\mathcal{D}_{\text{KL}}\left(\pi_\theta(\cdot \mid \mathbf{x}, \mathbf{y}_{<k}) \;\|\; \pi_\psi(\cdot \mid \mathbf{x}, \mathbf{y}_{<k})\right)\right], \tag{2}$$

where $\mathbf{y} = (y_1, \ldots, y_K)$ denotes a trajectory of K tokens sampled from the student policy, and $\mathcal{D}_{\text{KL}}(P \,\|\, Q) = \sum_{v\in\mathcal{V}} P(v)\log\frac{P(v)}{Q(v)}$ denotes the reverse KL divergence.

3.2 On-Policy Mid-Training Distillation (OPMD)

Applying On-Policy Distillation (OPD) to Continual Pre-Training (CPT) offers an appealing framework to overcome the rigid empirical matching of standard Next-Token Prediction (NTP), allowing the model to actively explore trajectory spaces and mitigate distribution shifts. However, directly extending token-level OPD to vast pre-training corpora introduces severe computational bottlenecks: performing autoregressive rollouts and continuous teacher forward passes across billions of unstructured tokens incurs prohibitive compute costs and severe window overlap redundancies. This raises a critical question: *How can we efficiently leverage vast, unstructured CPT corpora for on-policy distillation under bounded computational budgets?*

To address this challenge, we introduce *On-Policy Mid-Training Distillation* (OPMD). OPMD selectively triggers student trajectory rollouts at semantically critical trigger positions via *Anchor Token Identification* (ATI), converts preceding sequence context into prompts, and performs dense, token-level reverse KL distillation against a frozen teacher model.

Anchor Token Identification (ATI). Not all tokens in a continual pre-training sequence contribute equally to model distillation. Inspired by recent exploration strategies in reinforcement pre-training (Dong et al., 2025), ATI dynamically identifies critical trigger positions for student rollouts based on individual token-level uncertainty. Given an input text sequence $\mathbf{x} = (x_1, \ldots, x_T) \sim \mathcal{D}$, we utilize a lightweight, frozen proxy model π_ϕ (`DeepSeek-R1-Distill-Qwen-1.5B`) to scan the document. For long sequences exceeding the proxy's context limit, we process text using a sliding window of length 4096 with a stride of 2048, averaging predictive entropies across overlapping regions. The per-token uncertainty at index t is evaluated as:

$$\mathcal{H}_\phi(t) = -\sum_{v\in\mathcal{V}}\pi_\phi(v \mid \mathbf{x}_{<t})\log\pi_\phi(v \mid \mathbf{x}_{<t}). \tag{3}$$

To systematically investigate the effect of token uncertainty on mid-training distillation, we construct three distinct experimental arms based on per-token entropy distributions:

- *High-Entropy Arm:* We target positions with maximum predictive uncertainty to encourage active policy exploration, extracting the top 5% ($r = 0.05$) highest-entropy tokens per document:
$$\mathcal{S}_{\text{high}}(t) = \mathbb{I}\left(\mathcal{H}_\phi(t) \geq Q_{0.95}(\mathcal{H}_\phi)\right), \tag{4}$$
where $Q_\alpha(\mathcal{H}_\phi)$ denotes the α-quantile of the entropy values within a sequence, and $\mathbb{I}(\cdot)$ is the indicator function.

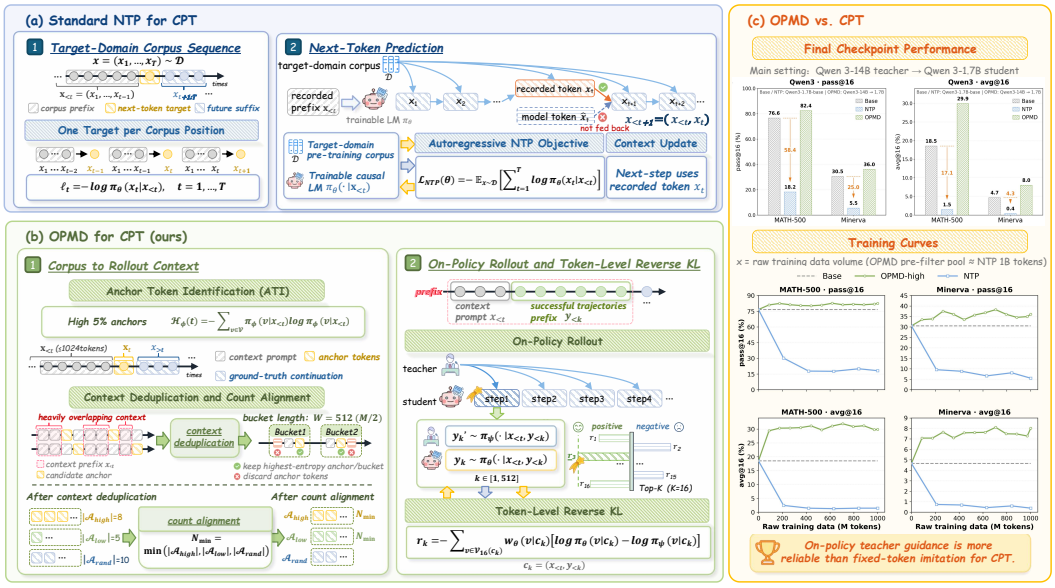


Figure 2: **Overview of the OPMD compared with standard NTP, along with training dynamics. (a) Standard NTP:** Rigid token-by-token imitation on specialized domain corpora leads to severe policy collapse. **(b) OPMD Pipeline:** (1) *Anchor Token Identification (ATI) & Alignment:* A proxy model identifies informative trigger tokens via predictive entropy, followed by context deduplication and strict budget alignment. (2) *On-Policy Rollout & Distillation:* The student generates continuations from prefix prompts and receives dense teacher guidance through a student-weighted Top-16 log-ratio reward at each rollout step. **(c) Performance Dynamics:** OPMD completely eliminates NTP policy collapse and achieves substantial, stable reasoning gains across benchmarks.

- *Low-Entropy Arm:* We extract the bottom 5% of tokens with the lowest individual entropy values via stable sorting, focusing on deterministic structural transitions:

$$\mathcal{S}_{\text{low}}(t) = \mathbb{I}\left(\mathcal{H}_\phi(t) \leq Q_{0.05}(\mathcal{H}_\phi)\right). \tag{5}$$

- *Stochastic Baseline:* Each token position is independently sampled as a candidate anchor with probability $p = 0.05$, using deterministic document-level seeds. The candidate-selection indicator is

$$\mathcal{S}_{\text{rand}}(t) \sim \text{Bernoulli}(p). \tag{6}$$

  As described below, these candidates undergo the same deduplication procedure as the other arms: the highest-entropy candidate is retained in each 512-token bucket before count alignment.

For each candidate anchor position t, the preceding prefix $\mathbf{x}_{<t} = (x_1, \ldots, x_{t-1})$ (truncated to a maximum context length $M = 1024$) serves as the conditioning context prompt from which the student generates its subsequent on-policy rollout.

Context Deduplication and Count Alignment. In dense text regions, candidate anchor tokens in close proximity share heavily overlapping context prefixes $\mathbf{x}_{<t}$. Triggering separate student rollouts on overlapping prefixes introduces severe computational redundancy without providing novel exploration signals. To resolve this, we apply a non-overlapping window deduplication procedure: for each candidate set independently, candidate anchors within the same document are partitioned into non-overlapping buckets of length $W = M/2$. Within each bucket, only the candidate anchor with the highest entropy is retained, while the remaining overlapping anchors are discarded. This entropy-based retention rule is applied independently to all three candidate sets, including the Low-Entropy and Stochastic arms. Thus, the Stochastic arm uses random candidate preselection, but its retained anchors are not uniformly random.

Furthermore, because spatial entropy distributions vary across text regions, deduplication yields different candidate set sizes $|\mathcal{A}_{\text{high}}|$, $|\mathcal{A}_{\text{low}}|$, and $|\mathcal{A}_{\text{rand}}|$. To ensure strict variable control and eliminate

sample-size confounding across ablation arms, we uniformly downsample the deduplicated anchor sets without replacement to match the minimum budget across all arms:

$$N_{\min} = \min\left(|\mathcal{A}_{\text{high}}|, |\mathcal{A}_{\text{low}}|, |\mathcal{A}_{\text{rand}}|\right). \tag{7}$$

This alignment guarantees that all three arms contain the same number of retained anchor prompts, $N_{\min}$.

On-Policy Rollout and Token-Level Reverse KL. For each validated anchor position $t \in \mathcal{A}(\mathbf{x})$, the preceding text $\mathbf{x}_{<t}$, truncated to the maximum context length $M = 1024$, serves as the prompt for an on-policy rollout. The student generates a single continuation $\mathbf{y} = (y_1, \ldots, y_K)$ of up to $K = 512$ tokens:

$$y_k \sim \pi_\theta\left(\cdot \mid \mathbf{x}_{<t}, \mathbf{y}_{<k}\right), \qquad k = 1, \ldots, K. \tag{8}$$

The frozen teacher π_ψ then evaluates the prefixes visited by the student. Thus, distillation guidance is provided on student-generated continuations rather than exclusively on the fixed token sequences in the pre-training corpus.

At each rollout step k, let $c_k = (\mathbf{x}_{<t}, \mathbf{y}_{<k})$, and let $\mathcal{V}_{16}(c_k)$ denote the student's Top-16 next-token candidates. We assign each candidate a weight proportional to its student probability,

$$w_\theta(v \mid c_k) = \frac{\pi_\theta(v \mid c_k)}{\sum_{u \in \mathcal{V}_{16}(c_k)} \pi_\theta(u \mid c_k)}. \tag{9}$$

The token-level distillation reward aggregates the teacher–student log-probability differences over these candidates:

$$r_k = -\sum_{v \in \mathcal{V}_{16}(c_k)} w_\theta(v \mid c_k)\left[\log \pi_\theta(v \mid c_k) - \log \pi_\psi(v \mid c_k)\right]. \tag{10}$$

This reward retains the student-weighted log ratio underlying reverse KL while restricting the comparison to a small set of high-probability candidates. The student probabilities are normalized within the Top-16 set, while the teacher log-probabilities retain their full-vocabulary values. It therefore provides a Top-16 approximation to the full-vocabulary distillation signal.

Overall Training Objective. Given the anchor prompts and student-generated continuations, OPMD aims to maximize the accumulated teacher-guided rewards. We express this intended distillation objective as

$$\mathcal{L}_{\text{OPMD}}(\theta) = -\mathbb{E}_{\substack{\mathbf{x}\sim\mathcal{D},\ t\sim\mathcal{A}(\mathbf{x}),\\ \mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x}_{<t})}}\left[\sum_{k=1}^{K} r_k\right]. \tag{11}$$

Here, the expectation covers corpus sequences, selected anchor positions, and continuations sampled from the evolving student policy. In implementation, we use the individual candidate contributions in Eq. (10), rather than the aggregated scalar r_k, as direct advantage signals within the veRL PPO framework. Equation (11) states the intended distillation target; PPO supplies the practical update mechanism. This allows OPMD to train on student-visited contexts using dense teacher feedback without outcome rewards or external verifiers.

4 Experiments

4.1 Experimental Settings

Training Data. We perform mid-training on a 1B-token subset of MegaMath-Web-Pro (Zhou et al., 2025). Per-token predictive uncertainty is calculated via a frozen proxy model (`DeepSeek-R1-Distill-Qwen-1.5B` (Guo et al., 2025a)) to construct three ablation arms: High-Entropy (top 5%), Low-Entropy (bottom 5%), and Stochastic Baseline (5% random candidate preselection followed by entropy-based deduplication). Each anchor token defines a preceding context prompt of up to 1024 tokens, targeting an on-policy rollout of up to 512 tokens. Following 512-token window deduplication, all arms are uniformly downsampled to an identical budget of 435,049 training sequences to ensure strict variable control across the three anchor-selection strategies.

Implementation Details. Built on `veRL` (Sheng et al., 2025) with `vLLM` (Kwon et al., 2023), we employ `Qwen3-1.7B-Base` (Yang et al., 2025) as the student model and a frozen

Table 1: **Main results on MATH-500 and Minerva** (n=16 sampling). Highlighted values and arrows indicate performance gains or losses relative to the `Qwen3-1.7B-Base` student model.

Methods	MATH-500		Minerva	
	pass@16	avg@16	pass@16	avg@16
Qwen3-1.7B-Base (Student)	76.6	18.5	30.5	4.7
Qwen3-4B-Base (Teacher 1)	84.8	21.3	34.9	5.6
Qwen3-14B-Base (Teacher 2)	94.2	68.6	51.5	24.0
Student NTP	**18.2**$^{\downarrow 58.4}$	**1.5**$^{\downarrow 17.0}$	**5.5**$^{\downarrow 25.0}$	**0.4**$^{\downarrow 4.3}$
Teacher 1 OPMD	**82.8**$^{\uparrow 6.2}$	**30.5**$^{\uparrow 12.0}$	**34.9**$^{\uparrow 4.4}$	**7.0**$^{\uparrow 2.3}$
Teacher 2 OPMD	**82.4**$^{\uparrow 5.8}$	**29.9**$^{\uparrow 11.4}$	**36.0**$^{\uparrow 5.5}$	**8.0**$^{\uparrow 3.4}$

`Qwen3-14B-Base` (Yang et al., 2025) as the teacher. The student generates a single trajectory per prompt ($n = 1$, temperature 1.0), supervised by a student-weighted Top-16 log-ratio reward and a reference KL penalty ($\beta = 0.005$). All OPMD runs use the same training hyperparameters: one epoch of AdamW optimization with a learning rate of 1×10^{-6} on 4–8 NVIDIA A100/A800 GPUs. The NTP baselines use a learning rate of 1×10^{-5}. Additional training and implementation details are provided in Appendix A.

Benchmarks. We evaluate models on mathematical reasoning tasks across MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) and Minerva (Lewkowycz et al., 2022). For each problem, we sample 16 candidate responses using stochastic decoding and report **pass@16** and **avg@16** via rule-based boxed answer matching. The first metric measures whether any of the 16 samples is correct, whereas the second reflects the average correctness of the sampled responses.

4.2 Main Results

We evaluate the effectiveness of On-Policy Mid-Training Distillation (OPMD) on mathematical reasoning benchmarks across various training paradigms, teacher model capacities, and intermediate checkpoints. Our empirical results highlight three key advantages:

(i) Superiority Over Standard NTP Training. As summarized in Table 1, standard Next-Token Prediction (NTP) suffers from substantial performance degradation on both math benchmarks during continual pre-training on domain-specific mathematical text. Throughout CPT, the NTP baseline's performance drops precipitously from the base model's initial 76.6% pass@16 down to 18.2% on MATH-500 (with avg@16 collapsing from 18.5% to 1.5%) and from 30.5% down to 5.5% pass@16 on Minerva (with avg@16 dropping from 4.7% to 0.4%) in our evaluation.

In stark contrast, OPMD improves benchmark performance under the evaluated training settings and unlocks substantial reasoning gains across both teacher configurations. Specifically, distilling with the compact `Qwen3-4B` teacher (Teacher 1) achieves the highest solve rates on MATH-500, pushing pass@16 to 82.8% (+6.2%) and avg@16 to 30.5% (+12.0%). Meanwhile, scaling to the `Qwen3-14B` teacher (Teacher 2) yields the strongest capacity on Minerva, reaching 36.0% pass@16 (+5.5%) and 8.0% avg@16 (+3.4%). The two metrics reveal different aspects of this improvement.

For example, with the 4B teacher on MATH-500, pass@16 increases by 6.2 percentage points relative to the base student, whereas avg@16 increases by 12.0 points. Thus, the gain is reflected not only in the set of problems solved at least once among 16 samples, but also in the frequency of correct responses across those samples. Overall, all OPMD variants consistently outperform the baseline student and avoid the degradation observed for our NTP baseline.

(ii) Scalability and Teacher Capacity Dynamics.

OPMD demonstrates notable scalability across teacher model capacities (Qwen3-4B-Base vs. Qwen3-14B-Base). As depicted in Figure 3, distilling the `Qwen3-1.7B` student with a compact 4B teacher yields closely comparable performance trajectories to those guided by the substantially larger 14B teacher across the entire training token regime (e.g., final MATH-500 pass@16 of 82.8% for 4B vs. 82.4% for 14B; final Minerva pass@16 of 34.9% for 4B vs. 36.0% for 14B). Note that while total step counts differ due to hardware cluster configurations (1699 steps on 4 GPUs for 4B vs. 1132

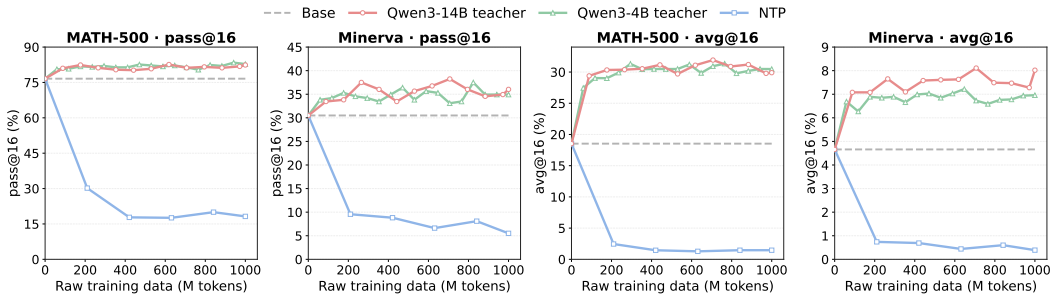


Figure 3: **Training dynamics of OPMD with different teacher scales versus standard NTP.** Performance trajectories (`pass@16` and `avg@16`) of the `Qwen3-1.7B` student on MATH-500 and Minerva are plotted against raw training token volume (in millions). NTP performance declines, whereas OPMD improves with either the Qwen3-4B or Qwen3-14B teacher. The two teacher settings show broadly comparable trajectories over the evaluated range.

steps on 6 GPUs for 14B), both settings reflect an equivalent overall token exposure. Two possible explanations for this plateau under matched token exposure are:

- *Student Capacity Ceiling:* The limited parameter capacity of a 1.7B student may constrain its ability to assimilate increasingly granular soft-target nuances beyond a certain threshold. The log-probabilities supplied by a 4B teacher for the top-16 candidates appear sufficient to provide effective directional supervision for a student of this scale, as reflected in the similar teacher-guided trajectories on both evaluated benchmarks. This offers a capacity-based explanation for the small performance differences between the two teacher settings.
- *State-Space Distributional Alignment:* Because trajectory rollouts are generated on-policy by the student ($\mathbf{y} \sim \pi_\theta$), evaluated states remain anchored within the student's policy distribution. Over these visited contexts, the 4B and 14B teachers may assign similar relative log-probabilities to influential candidates in the student's Top-16 set, yielding broadly congruent Reverse KL update directions. Similar supervision distributions imply similar updates on the contexts that the student actually visits.

These empirical findings underscore a practical benefit of OPMD regarding **teacher compute efficiency**: strong mid-training distillation gains can be realized with a relatively compact teacher, substantially reducing inference and memory overhead during optimization without meaningfully sacrificing target performance for the Qwen3-1.7B student on the two tested benchmarks.

(iii) Exceptional Sample Efficiency and Training Stability. Across all OPMD variants, the student model achieves remarkable data efficiency, capturing substantial distillation gains within bounded optimization steps. With the 14B teacher, the student's MATH-500 avg@16 rises from 18.5% at initialization to 29.4% at step 100, the first evaluated checkpoint, while Minerva avg@16 rises from 4.7% to 7.1%. These gains persist despite fluctuations at later checkpoints: at the final checkpoint of this training run, avg@16 reaches 29.9% on MATH-500 and 8.0% on Minerva (Appendix Table 2). The corresponding pass@16 scores also remain above the base model throughout the evaluated run. These results suggest that on-policy distillation over anchor-triggered contexts can sustain its early improvements throughout the evaluated training budget, including at later checkpoints.

4.3 Ablation Studies and Analysis

Effectiveness of Different ATI Strategies.

To isolate the effect of anchor token selection quality, all experimental arms are strictly controlled under an identical training budget of 435,049 samples, aligned via context-aware deduplication over non-overlapping 512-token windows. We systematically evaluate three selection strategies: High-Entropy, Low-Entropy, and Stochastic Baseline (Rand). As illustrated in Figure 4, empirical results yield three key insights about peak solution density, low-entropy anchors, and performance under random preselection. Because all arms use the same sample budget, the comparisons isolate the effect of where rollouts begin rather than differences in training volume:

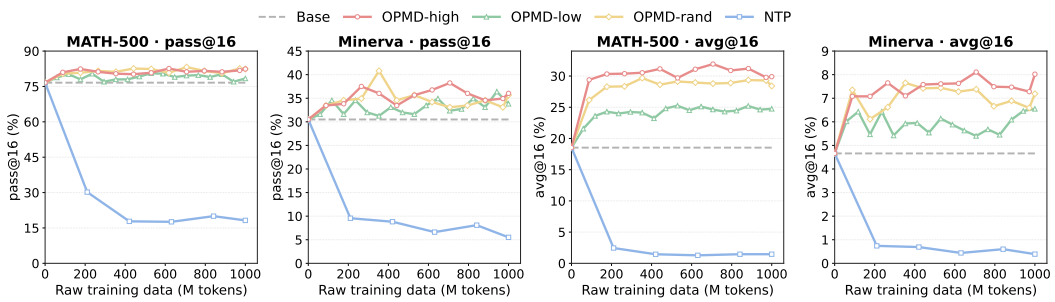


Figure 4: **Training dynamics of Anchor Token Identification (ATI) strategies.** MATH-500 and Minerva `pass@16` and `avg@16` for the Qwen3-1.7B student across all three ATI strategies are plotted against raw training token volume (millions). In the evaluated setting, OPMD variants remain above the base model throughout training, whereas the NTP baseline degrades rapidly. Among these arms, OPMD-high attains the highest observed peak `avg@16` on both benchmarks.

- *High-Entropy Candidate Preselection and Peak Solution Density:* Distilling over high-entropy token positions achieves the highest peak solution consistency (*avg@16*) across benchmarks, reaching **31.9%** avg@16 on MATH-500 and **8.1%** avg@16 on Minerva (both at step 800). High entropy appears to mark local epistemic uncertainty and critical logical branching points in pre-training corpora. Triggering on-policy student rollouts at these decision junctions provides rich teacher supervision, contributing to the higher avg@16 observed in this setting. The peak values show that high-entropy anchors improve solution consistency on both benchmarks while using the same sample budget as the other strategies.
- *Limitations of Low-Entropy Anchors:* Conversely, the Low-Entropy arm yields lower overall gains among OPMD variants, peaking at 25.3% avg@16 on MATH-500 (step 900) and 6.6% on Minerva (step 1699). Low-entropy positions predominantly correspond to highly deterministic structural or syntactic continuations where the next token is easily predictable. Distilling over such confident context offers limited active exploration value to the student policy. The lower peaks on both benchmarks are consistent with this limitation of low-entropy anchors, with the same ordering observed on MATH-500 and Minerva.
- *Framework Robustness and Stochastic Baseline Performance:* Remarkably, the Stochastic (Rand) baseline demonstrates strong instance-level exploration, achieving peak pass@16 scores of **83.2%** on MATH-500 (step 800) and **40.8%** on Minerva (step 400). This indicates that the core OPMD paradigm—converting static CPT context into active prompt-rollout triggers—is inherently robust to anchor selection. Nevertheless, High-Entropy selection generally exhibits superior average solve rate density (*avg@16*), suggesting greater trajectory stability over extended training. The two metrics thus distinguish the stochastic arm's peak pass@16 from the high-entropy arm's stronger avg@16.

5 Conclusion

We studied continual pre-training on a fixed mathematical corpus and observed that the same NTP training recipe improves Llama-3.2-3B but degrades Qwen3-1.7B-Base on the evaluated reasoning benchmarks. We introduced On-Policy Mid-Training Distillation (OPMD), which turns selected corpus prefixes into rollout prompts and trains on student-generated continuations using dense, token-level feedback from a frozen teacher model. On MATH-500 and Minerva, OPMD avoids the regression observed with NTP and improves the Qwen student over its base model on both pass@16 and avg@16 across the two evaluated teacher configurations. In this setting, 4B and 14B teachers yield comparable gains on both mathematical reasoning benchmarks, while high-entropy candidate preselection achieves the highest observed peak avg@16 among the tested selection procedures. These results suggest that on-policy distillation is a promising objective for mathematical mid-training. Evaluation across additional domains and with uniformly random final anchors will help establish the generality of these findings and isolate the effect of anchor selection.

AI USE STATEMENT

We used LLM-based tools to assist with drafting and polishing the manuscript and preparing figures based on our experimental data. The tools did not generate or alter the reported experimental data or numerical results. The authors reviewed and verified the AI-assisted material and take full responsibility for the accuracy and integrity of the paper.

REPRODUCIBILITY STATEMENT

Sections 4 and Appendix A describe the data, anchor construction, training, and evaluation procedures. Appendix A.2 identifies the public generation and grading scripts, and Appendix B reports checkpoint results.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5be69a584901a26c521c2b51e40a4c20-Paper-Conference.pdf.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, Yankai Lin, Ruihua Song, Jun Xu, Xu Chen, Rui Yan, Zhewei Wei, Di Hu, Wenbing Huang, and Ji-Rong Wen. Towards effective and efficient continual pre-training of large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5779–5795, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.289. URL https://aclanthology.org/2025.acl-long.289/.

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training, 2025. URL https://arxiv.org/abs/2506.08007.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *Int. J. Comput. Vision*, 129(6):1789–1819, June 2021. ISSN 0920-5691. doi: 10.1007/s11263-021-01453-z. URL https://doi.org/10.1007/s11263-021-01453-z.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8ac015d409635f196f9e3e9dcfb9a94e-Paper-Conference.pdf.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 17842–17859, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4db16435e3a5a2ef3fc39b8f0d12498d-Paper-Conference.pdf.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025a.

Yiduo Guo, Jie Fu, Huishuai Zhang, and Dongyan Zhao. Efficient domain continual pretraining by mitigating the stability gap. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32850–32870, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1578. URL https://aclanthology.org/2025.acl-long.1578/.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=m_GDIItaI3o.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026. URL https://arxiv.org/abs/2604.13016.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Ruiqi Wang, Zezhou Yang, Cuiyun Gao, Xin Xia, and Qing Liao. An empirical study of knowledge distillation for code understanding tasks, 2025. URL https://arxiv.org/abs/2508.15423.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818. Curran Associates, Inc., 2023a. doi: 10.52202/075280-3059. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/dcba6be91359358c2355cd920da3fcbd-Paper-Conference.pdf.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227. Curran Associates, Inc., 2023b. doi: 10.52202/075280-1482. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6b9aa8f418bde2840d5f4ab7a02f663b-Paper-Conference.pdf.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yizhou Ying, Geng Zhang, Cui Danxin, Chengyu Du, Guanglei Yue, Sihang Jiang, Jiaqing Liang, Yifei Fu, Hailin Hu, and Yanghua Xiao. Data-efficient selection via grammatical complexity in continual pre-training of domain-specific LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 22055–22069, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1121. URL https://aclanthology.org/2025.emnlp-main.1121/.

Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P. Xing. Megamath: Pushing the limits of open math corpora, 2025. URL https://arxiv.org/abs/2504.02807.

APPENDIX

A Additional Implementation Details **13**

A.1 Training Corpus and Anchor Construction 13

A.2 Distillation Implementation 13

A.3 Evaluation Protocol 14

B Additional Checkpoint Results **14**

B.1 Teacher-Scale Comparison 14

B.2 NTP Checkpoints 14

B.3 Anchor-Selection Comparison 14

A ADDITIONAL IMPLEMENTATION DETAILS

A.1 TRAINING CORPUS AND ANCHOR CONSTRUCTION

All NTP and OPMD experiments use the same sampled set of original documents from the 1B-token MegaMath-Web-Pro subset described in Section 4.1. The methods use these documents differently: NTP learns from the recorded continuations, whereas OPMD converts selected document prefixes into prompts and trains on student-generated continuations. Thus, sharing source documents does not imply that the methods receive identical optimization examples.

For anchor construction, a frozen `DeepSeek-R1-Distill-Qwen-1.5B` proxy evaluates predictive entropy along each document. As described in Section 3.2, long documents are processed with windows of 4096 tokens and a stride of 2048 tokens; entropy estimates in overlapping regions are averaged. High- and Low-Entropy candidates correspond to the highest and lowest 5% of token positions within a document, respectively. For the Stochastic arm, each token position is independently preselected with probability 0.05 using deterministic document-level seeds.

Deduplication is performed separately for each arm. Within each non-overlapping 512-token bucket, we retain the candidate with the highest proxy entropy. Consequently, final anchors in the Stochastic arm are not a uniformly random sample of token positions. We then uniformly downsample the three deduplicated sets without replacement to 435,049 anchor prompts per arm. Each prompt consists of up to 1024 tokens preceding its selected anchor; the anchor token itself is not included in the prompt.

A.2 DISTILLATION IMPLEMENTATION

The OPMD student is `Qwen3-1.7B-Base`. We evaluate `Qwen3-4B-Base` and `Qwen3-14B-Base` as frozen teachers. For each anchor prompt, the student generates one on-policy continuation of at most 512 tokens. At each generated position, the distillation signal uses the student's top-16 next-token candidates. Student probabilities are normalized within this candidate set, while teacher log-probabilities are taken from the teacher's full-vocabulary distribution. The individual candidate contributions are used as advantage signals in the PPO-based implementation, as described in Section 3.2.

The reported training configuration uses one epoch, a learning rate of 1×10^{-6}, and a reference-policy KL coefficient of $\beta = 0.005$. The different optimizer step counts in the checkpoint tables should not be interpreted as different numbers of source documents. Comparisons across training runs are shown against raw training token volume in the main-text figures.

A.3 Evaluation Protocol

We use the `gen_vllm.py` generation script and `grade.py` grading script from the public Rethinking OPD implementation.[1] For each problem in MATH-500 or Minerva, we generate 16 responses and score each response using the same grading procedure. Let $c_{ij} \in \{0, 1\}$ indicate whether response j to problem i is graded correct, and let N be the number of problems. We report

$$\text{pass@16} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\sum_{j=1}^{16} c_{ij} > 0\right], \tag{12}$$

$$\text{avg@16} = \frac{1}{16N} \sum_{i=1}^{N} \sum_{j=1}^{16} c_{ij}. \tag{13}$$

We multiply both quantities by 100 when reporting them as percentages. Pass@16 measures the fraction of problems for which at least one of the 16 responses is correct; avg@16 measures the average correctness of an individual response over the same samples.

B Additional Checkpoint Results

The main text presents training trajectories against raw training token volume. Here we report the recorded checkpoint scores underlying those trajectories. For each benchmark, pass@16 is the percentage of problems with at least one correct response among 16 samples, while avg@16 is the average correctness across the 16 responses. All scores are percentages rounded to one decimal place. Checkpoints from the same run are successive evaluations, not independent training repetitions. Since optimizer step counts differ across runs, step numbers should not be used as a direct measure of matched training exposure.

B.1 Teacher-Scale Comparison

Table 2 compares High-Entropy OPMD runs using Qwen3-4B-Base and Qwen3-14B-Base as teachers. Both runs start from the same Qwen3-1.7B-Base student.

B.2 NTP Checkpoints

Table 3 reports the available Qwen3-1.7B NTP checkpoints and the initial and final results of the Llama-3.2-3B NTP pilot experiment. The two NTP experiments use the same sampled set of original source documents.

B.3 Anchor-Selection Comparison

Table 4 lists all three anchor-selection trajectories, including the High-Entropy results from Table 2. Figure 4 compares the arms by raw training data volume.

[1] https://github.com/Thinking-Space/Rethinking-OPD

Table 2: High-Entropy OPMD checkpoint results with 4B and 14B teachers. The initial student is Qwen3-1.7B-Base. Each metric is computed from 16 responses per question and reported in %.

Checkpoint	MATH-500		Minerva	
	pass@16	avg@16	pass@16	avg@16
Teacher: Qwen3-4B-Base				
Student (initial)	76.6	18.5	30.5	4.7
Steps 100	80.6	27.5	33.8	6.7
Steps 200	80.8	29.0	34.2	6.3
Steps 300	81.8	29.0	35.3	6.9
Steps 400	81.6	29.9	34.6	6.9
Steps 500	82.0	31.3	34.2	6.9
Steps 600	81.4	30.5	33.5	6.7
Steps 700	81.4	30.5	34.9	7.0
Steps 800	82.6	30.5	36.4	7.0
Steps 900	82.2	30.5	33.8	6.9
Steps 1000	81.8	31.2	35.7	7.0
Steps 1100	82.2	29.9	35.3	7.2
Steps 1200	81.4	30.9	33.1	6.7
Steps 1300	80.4	31.3	33.5	6.6
Steps 1400	82.4	29.8	37.5	6.8
Steps 1500	82.0	30.2	34.9	6.8
Steps 1600	83.4	30.5	34.9	6.9
Steps 1699 (final)	82.8	30.5	34.9	7.0
Teacher: Qwen3-14B-Base				
Student (initial)	76.6	18.5	30.5	4.7
Steps 100	81.0	29.4	33.5	7.1
Steps 200	82.4	30.3	33.8	7.1
Steps 300	81.2	30.4	37.5	7.7
Steps 400	80.4	30.5	36.0	7.1
Steps 500	80.2	31.2	33.5	7.6
Steps 600	80.8	29.7	35.7	7.6
Steps 700	82.6	31.1	36.8	7.6
Steps 800	81.2	31.9	38.2	8.1
Steps 900	81.6	30.9	36.0	7.5
Steps 1000	81.2	31.2	34.6	7.5
Steps 1100	81.8	29.8	34.9	7.3
Steps 1132 (final)	82.4	29.9	36.0	8.0

Table 3: Available NTP checkpoint results on MATH-500 and Minerva. Each metric is computed from 16 responses per question and reported in %.

Model and checkpoint	MATH-500		Minerva	
	pass@16	avg@16	pass@16	avg@16
Qwen3-1.7B-Base (initial)	76.6	18.5	30.5	4.7
Qwen3-1.7B NTP, step 400	30.2	2.5	9.6	0.7
Qwen3-1.7B NTP, step 800	17.8	1.5	8.8	0.7
Qwen3-1.7B NTP, step 1200	17.6	1.3	6.6	0.4
Qwen3-1.7B NTP, step 1600	20.0	1.5	8.1	0.6
Qwen3-1.7B NTP, step 1902 (final)	18.2	1.5	5.5	0.4
Llama-3.2-3B-Base (initial)	11.2	1.1	7.3	1.0
Llama-3.2-3B NTP (final)	37.0	6.2	17.3	2.4

Table 4: Checkpoint trajectories for three anchor-selection arms with the Qwen3-14B-Base teacher. Results within each arm are listed in training order. Each metric is computed from 16 responses per question and reported in %. Step numbers identify checkpoints within a run and are not matched training-token budgets across arms.

Checkpoint	MATH-500		Minerva	
	pass@16	avg@16	pass@16	avg@16
High-Entropy				
Student (initial)	76.6	18.5	30.5	4.7
Steps 100	81.0	29.4	33.5	7.1
Steps 200	82.4	30.3	33.8	7.1
Steps 300	81.2	30.4	37.5	7.7
Steps 400	80.4	30.5	36.0	7.1
Steps 500	80.2	31.2	33.5	7.6
Steps 600	80.8	29.7	35.7	7.6
Steps 700	82.6	31.1	36.8	7.6
Steps 800	81.2	31.9	38.2	8.1
Steps 900	81.6	30.9	36.0	7.5
Steps 1000	81.2	31.2	34.6	7.5
Steps 1100	81.8	29.8	34.9	7.3
Steps 1132 (final)	82.4	29.9	36.0	8.0
Low-Entropy				
Student (initial)	76.6	18.5	30.5	4.7
Steps 100	78.8	21.6	31.6	6.0
Steps 200	80.2	23.6	34.6	6.4
Steps 300	78.0	24.3	31.6	5.5
Steps 400	80.4	24.0	34.6	6.4
Steps 500	77.0	24.3	32.0	5.4
Steps 600	78.0	24.1	31.3	5.9
Steps 700	78.0	23.2	33.1	6.0
Steps 800	79.2	24.8	32.0	5.5
Steps 900	80.6	25.3	31.6	6.1
Steps 1000	80.4	24.5	33.5	5.9
Steps 1100	79.0	25.1	34.9	5.6
Steps 1200	79.6	24.6	32.4	5.4
Steps 1300	79.8	24.3	32.7	5.7
Steps 1400	79.0	24.5	34.9	5.5
Steps 1500	80.2	25.2	33.1	6.1
Steps 1600	77.0	24.6	36.4	6.5
Steps 1699 (final)	78.4	24.8	33.8	6.6
Stochastic Candidate Preselection				
Student (initial)	76.6	18.5	30.5	4.7
Steps 100	80.4	26.2	33.5	7.4
Steps 200	80.8	28.3	34.6	6.1
Steps 300	81.6	28.4	34.9	6.6
Steps 400	81.2	29.7	40.8	7.7
Steps 500	82.6	28.6	34.6	7.4
Steps 600	82.4	29.1	35.7	7.4
Steps 700	80.8	29.0	34.2	7.3
Steps 800	83.2	28.8	33.1	7.4
Steps 900	81.6	28.9	33.5	6.7
Steps 1000	80.6	29.4	34.6	6.9
Steps 1100	82.6	29.3	33.1	6.6
Steps 1132 (final)	82.4	28.4	35.7	7.2