ON-POLICY ATTENTION LINEARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid transformer architectures that replace most softmax attention layers with linear attention offer transformer-level quality at a fraction of the memory cost. Rather than pretraining such models, a growing body of work distills them from already trained full-attention transformers. However, these distilled models often collapse on long-context retrieval and reasoning tasks, particularly when operating in thinking mode, where the efficiency gains of hybrid architectures matter most. Since linear attention layers must compress context into a fixed-size state, their errors compound over long sequences. As off-policy distillation never teaches the student model to recover from this drift, tasks that necessitate longer sequence lengths become especially challenging. We introduce **O**n-**P**olicy **A**ttention **L**inearization (OPAL) in which the hybrid attention student samples its own long-context trajectories and receives dense supervision from the frozen full-attention teacher. Applying OPAL to Qwen3-4B and MiMo-7B-RL-0530, we recover 87–94% of full-attention performance on commonsense reasoning, 100% on needle-in-a-haystack (NIAH) retrieval, and 83–93% on mathematical reasoning with only 3B training tokens. We achieve these results without supervised fine-tuning (SFT) or reinforcement learning with verifiable rewards (RLVR). Compared with the strongest prior linearization method, which recovers 68% of its teacher's retrieval performance and 21.6% absolute average mathematical reasoning accuracy, OPAL fully recovers retrieval and achieves 67.6–72.2% on math reasoning.

1 INTRODUCTION

Attention-based large language models (LLMs) have become the de facto architecture for complex reasoning tasks (Vaswani et al., 2017; OpenAI, 2026; Guo, 2025). A primary drawback of this architecture is that the traditional softmax-based implementation of attention, often referred to as full attention, is quadratic in complexity as each token "attends" to all prior tokens in the sequence. For long-context reasoning tasks, this can be prohibitively expensive. The KV cache, which maintains the key and value projections of every token, grows linearly with the number of tokens, ultimately dominating overall memory costs (Pope et al., 2023; Kwon et al., 2023). To address the quadratic complexity of full attention, prior literature has proposed linear attention variants that maintain a constant-sized recurrent state rather than a growing KV cache, making per-token compute and memory independent of sequence length (Katharopoulos et al., 2020; Gu & Dao, 2024; Sun et al., 2023; Yang et al., 2024; 2025c). Recently, open-source models have begun to incorporate hybrid attention where a small fraction of the transformer blocks maintain full-attention layers, but a majority of the transformer blocks utilize a linear attention variant, such as Mamba or Gated DeltaNet (GDN) (Ren et al., 2025; Lenz et al., 2025; Team, 2026; Team et al., 2026). These hybrid models aim to balance the efficiency gains of linear attention with the long-context recall abilities of full attention.

Recent work aims to distill linear attention or hybrid attention models from full-attention teachers, making it possible to create deployment-friendly versions of full-attention transformers without the exorbitant costs associated with pretraining (Zhang et al., 2025; Goldstein et al., 2025; Bick et al., 2025; Yang et al., 2025b; Chen et al., 2026; Lan et al., 2026; Li et al., 2026). Such methods generally proceed in two stages. First, the student's linear attention layers are initialized from the teacher's full-attention weights and trained to reproduce its attention hidden states. Second, the whole student model is fine-tuned to match the teacher's next-token distribution, typically using a KL objective on pretraining-like data. Prior to these two stages, an initial stage is sometimes included to determine which layers are most suitable for linear attention conversion (Zhou et al., 2026; Bick et al., 2026).

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.

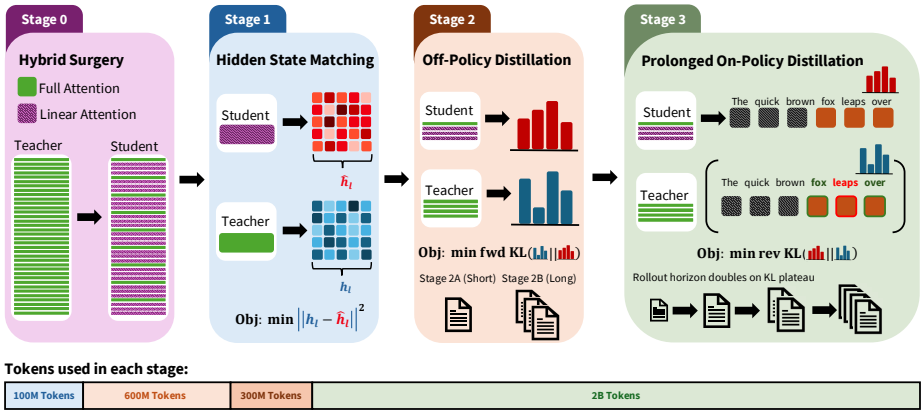


Figure 1: **Overview of OPAL.** We convert a full attention teacher into a hybrid student by replacing 75% of attention layers with linear attention. We recover the teacher's capabilities through hidden-state alignment, short- and long-context off-policy distillation, and prolonged on-policy distillation.

However, while prior works benchmark their distilled linear attention or hybrid attention variants on commonsense/likelihood suites, the efficiency advantage of linear and hybrid attention only materializes when processing long inputs or generating many tokens, where using the recurrent state eliminates KV cache memory pressure. Thus, the settings that motivate linearization are long-context retrieval and multi-step reasoning, in which models must locate and process information across thousands of tokens. On precisely these tasks, we find that existing distilled models fall far short of their full attention teachers, even when they perform well on likelihood benchmarks.

We introduce **O**n-**P**olicy **A**ttention **L**inearization (OPAL) a method for converting pretrained full-attention models into efficient hybrid attention models while preserving their long-context capabilities. OPAL retains the standard two-stage recipe of prior work and appends a third stage of on-policy distillation (OPD) (Agarwal et al., 2024; Lu & Lab, 2025) in which the student is trained on sequences it generates rather than teacher-forced contexts. OPAL proceeds in three key stages. First, we initialize the hybrid student's linear attention layers from the teacher's full-attention weights and align their hidden states. Next, we distill the hybrid student against the teacher's next-token distribution on a fixed corpus. Finally, the student samples its own continuations, the teacher scores every token, and the student minimizes the divergence from the teacher along its own trajectories (Figure 1). We improve the efficacy of the third stage of this pipeline, the OPD stage, in three ways.

- Training begins with short student rollouts and doubles their length each time the teacher-student KL plateaus. In this way, the student progressively improves on longer horizons.
- We run OPD as a prolonged continual-learning-style scaling ladder using warmup-stable-decay (WSD) for the learning rate schedule (Hu et al., 2024; Wen et al., 2025). This lets us extend training from 200M to 2B OPD tokens without restarting and yields checkpoints for measuring how reasoning recovers with on-policy compute.
- We apply a difficulty curriculum over the reasoning prompts used for rollouts, introducing harder tasks as the student's accuracy on easier ones saturates over the course of the OPD checkpoints.

We apply OPAL to Qwen3-4B and MiMo-7B-RL-0530 and distill each into a hybrid student with 75% of its full-attention layers replaced by GDN. With 2B OPD tokens, both students match their full-attention teachers on retrieval at every context length up to 32K, raise MATH-500 with thinking to 92% and 93.8% against teacher scores of 96.4% and 97%, and recover 92.3% and 100% of teacher pass@8 on AIME'24. Across six prior distillation methods, the best baseline retains 68% of its full-attention teacher's retrieval accuracy and 66.4% of its reasoning average, while OPAL students retain 100% of retrieval and 83% and 93% reasoning. In absolute terms, OPAL achieves a 67.6–72.2% reasoning average, over 3× the 21.6% of the strongest prior linearized model. Notably, both full-attention teacher models we consider are known to have undergone extensive RL post-training, yet our full attention to GDN conversion pipeline is able to recover much of their performance. Our results show that a trained transformer's retrieval and reasoning abilities can survive a late architectural conversion and that performance can be cheaply and efficiently recovered.

2 Problem Setup and Motivation

Attention Architectures and Hybrid Transformers. Given queries, keys, and values $q_t, k_t, v_t \in \mathbb{R}^d$ at token t and with per-head attention dimension d, a softmax attention layer computes

$$o_t \;=\; \sum_{i\le t} \frac{\exp\!\left(q_t^\top k_i/\sqrt{d}\right)}{\sum_{j\le t}\exp\!\left(q_t^\top k_j/\sqrt{d}\right)}\, v_i. \tag{1}$$

To compute softmax attention, the layer must maintain a history of all previous keys and values in a KV cache. Consequently, the KV cache and per-token compute grow linearly with sequence length. In contrast, linear attention replaces the exponential kernel with a feature map $\phi : \mathbb{R}^d \to \mathbb{R}^{d_\phi}$, so that $\exp(q^\top k/\sqrt{d}) \approx \phi(q)^\top \phi(k)$. Dropping the normalization from Equation 1, we get the following

$$o_t \;=\; \sum_{i\le t} \big(\phi(q_t)^\top \phi(k_i)\big)\, v_i \;=\; \sum_{i\le t} v_i\, \phi(k_i)^\top \phi(q_t) \;=\; \underbrace{\left(\sum_{i\le t} v_i\, \phi(k_i)^\top\right)}_{S_t} \phi(q_t). \tag{2}$$

Therefore, linear attention compresses the prefix into a recurrent state $S_t \in \mathbb{R}^{d\times d_\phi}$ so its per-token compute and state memory remain $O(1)$ with respect to sequence length. Each token attends to the prefix through S_t rather than through the exact keys and values. Linear attention variants differ in how they write to this state. Equation 2 implies an additive update $S_t = S_{t-1} + v_t\phi(k_t)^\top$, whereas the more expressive GDN layer updates S_t according to the delta rule (Yang et al., 2024) as follows

$$S_t \;=\; \alpha_t\, S_{t-1}\left(I - \beta_t\, k_t k_t^\top\right) + \beta_t\, v_t k_t^\top, \tag{3}$$

where $\alpha_t \in (0,1)$ is a data-dependent decay term and $\beta_t \in (0,1)$ controls the update strength. For simplicity, we consider the common case $d_\phi = d$ and absorb the transformation into the notation q_t and k_t. The combination gives the layer both global forgetting and targeted overwriting, allowing GDN to perform comparably to full attention. Hybrid attention architectures interleave a small number of softmax attention layers among linear attention layers. Some open-weight models, such as the Qwen3.5 family (Qwen Team, 2026a), Kimi K3 (Team et al., 2026), and Nemotron 3 Nano NVIDIA et al. (2025) have begun adopting this model design to better balance efficiency and performance.

Distillation for Linearization. Rather than pretraining linear or hybrid models from scratch, many works distill (Hinton et al., 2015; Kim & Rush, 2016) an existing full-attention transformer into a linear or hybrid student (Zhang et al., 2025; Goldstein et al., 2025; Li et al., 2026; Lan et al., 2026). These methods produce efficient variants of trained models at a small fraction of their original training cost. In practice, these methods follow a two-stage distillation recipe; they first align the outputs of each converted linear attention layer to the full-attention layer it replaces and then fine-tune the student against the teacher's next-token distribution. Let p_T denote the teacher distribution, p_θ the student distribution, and $\mathcal{D}$ a fixed text corpus. The forward KL loss is given by

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{x\sim\mathcal{D}} \sum_t D_{\mathrm{KL}}(p_T(\cdot \mid x_{<t}) \,\|\, p_\theta(\cdot \mid x_{<t}))\,, \text{ where } D_{\mathrm{KL}}(p_T \,\|\, p_\theta) \;=\; \mathbb{E}_{y\sim p_T}\left[\log \frac{p_T(y)}{p_\theta(y)}\right]. \tag{4}$$

We call this objective *off-policy* because the prefixes $x_{<t}$ on which the student is supervised are drawn from $\mathcal{D}$ and never from the student itself. Note that $\mathcal{D}$ may consist of teacher generations, which is still off-policy since the prefixes are also not the student's own traces. Ultimately, in either case, Equation 4 provides no direct supervision on prefixes sampled from the student's own policy. Prior linearization methods, such as RADLADS (Goldstein et al., 2025), Zebra-Llama (Yang et al., 2025b), and HALO (Chen et al., 2026), all optimize Equation 4 or a close variant on a fixed corpus, even if they differ in their initialization of converted attention layers or the way they schedule the conversion in their pipelines.

Distilled Hybrid Students on Long-Context Tasks. Consider a transformer with L layers and per-head dimension d applied to a length n sequence. For a full-attention model, cache memory is

$$M_{\mathrm{FA}}(n) = O(Lnd), \tag{5}$$

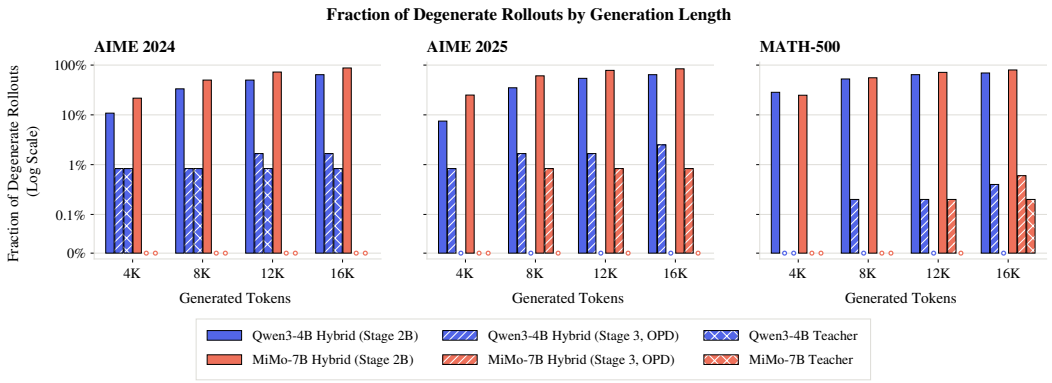


Figure 2: Degenerate rollouts during student decoding. Off-policy distillation aligns the student on fixed prefixes, but provides no signal on the student's own trajectories. Student rollouts can collapse into degenerate rollouts such as repetitive text or unrelated prose. Open circles indicate 0%.

omitting attention heads. Now consider a hybrid model in which only L_{FA} layers retain full attention. Since linear attention layers store a fixed-size recurrent state S_t, its memory scales as

$$M_{\text{Hybrid}}(n) = O(L_{\text{FA}} n d) + O((L - L_{\text{FA}})d^2). \tag{6}$$

The first term grows with context length, while the second remains fixed. As n increases, the memory savings from replacing full-attention layers become increasingly significant.

However, strong short-context performance after linearization does not guarantee long-context robustness. As shown in Figure 2, distilled hybrid students that are trained with attention matching and off-policy distillation can diverge from their teachers over long generations and fall into repetitive or degenerate trajectories. For linearized models, once an error changes the generated prefix, it also changes the recurrent state S_t that compresses that history. This ultimately affects every subsequent prediction. Because off-policy distillation in Equation 4 only supervises fixed prefixes from $\mathcal{D}$, the student is never explicitly trained to recover from states induced by its own mistakes. Over long horizons, these deviations can compound. Our goal is to linearize trained transformers without sacrificing the long-context retrieval and reasoning abilities that make efficient decoding valuable in the first place. This requires training the student to remain well-behaved on its own trajectories.

3 Proposed Method

OPAL converts a trained full-attention model into an efficient hybrid model while preserving its learned representations and behavior. Starting from the full-attention teacher, we construct a student parameterized by θ by retaining every fourth full-attention layer and replacing the remaining layers with GDN layers. For the 36-layer models considered in this work, the resulting hybrid architecture contains 9 interleaved full attention layers and 27 GDN layers (Figure 1). All non-attention components, such as the embeddings, MLPs, normalization layers, and output head, are inherited directly from the teacher. Compatible full-attention projections are used to initialize the corresponding GDN projections. The converted layers also introduce the GDN-specific forget and update gates α and β, an output gate, and a short causal convolution (Yang et al., 2024).

Architectural conversion substantially affects the model's representations and next-token distribution. To address this issue, we recover the teacher through the three training stages illustrated in Figure 1. We align the representations produced by the converted layers, optimize θ to match the teacher on fixed short- and long-context sequences, and train on student sampled trajectories.

Stage 1: Hidden State Matching. We first align each converted GDN layer independently with its corresponding teacher layer. For a sequence $\mathbf{x} = (x_1, \ldots, x_T) \sim \mathcal{D}$, let $h_{\ell,t}$ and $\hat{h}_{\ell,t}$ denote the teacher and student hidden states at layer ℓ and position t. We optimize the following

$$\mathcal{L}_{\text{match}}(\theta) = \frac{1}{|L_{\text{GDN}}|} \sum_{\ell \in L_{\text{GDN}}} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \left[\frac{1}{T} \sum_{t=1}^{T} \left\| h_{\ell,t} - \hat{h}_{\ell,t} \right\|_2^2 \right]. \tag{7}$$

This loss isolates the architectural mismatch introduced by replacing full attention with GDN. The matching stage ensures the model outputs coherent, albeit inaccurate and unrelated, text. During this stage, only the parameters of the converted GDN layers are updated. The retained full-attention layers and all other model parameters are frozen (equivalently, assigned 0 learning rate).

Stage 2: Off-Policy Distillation. Hidden state matching trains each converted layer independently but does not ensure that the entire hybrid model reproduces the teacher's next-token distribution. In Stage 2, we train all of the hybrid student's model weights with a short off-policy distillation stage. We use the objective in Equation 4, matching the student's predictive distribution to the teacher's at each position along ground-truth corpus sequences. This stage aims to remedy errors introduced by composing the individually aligned GDN layers. We first distill on short sequences 4K in length (Stage 2A) before extending to longer sequences 32K in length (Stage 2B) as a curriculum for improving on longer contexts. We train the entire hybrid model with two different step sizes, using a learning rate for retained full-attention layers that is $10\times$ smaller than for GDN layers.

Despite training on different-length sequences during Stage 2, we view this stage primarily as a warm start for OPD rather than the mechanism responsible for long-context recovery. Because every training prefix remains drawn from a fixed corpus, the student is never exposed to states induced by its own predictions. Therefore, off-policy distillation can provide a well-calibrated starting point, but it does not directly address the compounding state drift that arises during long rollouts.

Stage 3: Prolonged On-Policy Distillation. While prior works have considered analogues of Stages 1 and 2, our novel addition to the linearization pipeline is a stage of prolonged OPD. As mentioned, during generation, the student must condition on its own previous predictions. Thus, prolonged OPD is the only stage aimed at addressing the mismatch between prior objectives and model inference. Given a prompt $\mathbf{x}$, the student samples a continuation $\mathbf{y} = (y_1, \ldots, y_H)$ from its current policy. The frozen teacher is then applied to this trajectory, and we minimize the token-level reverse KL

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\, \mathbf{y}\sim p_\theta(\cdot|\mathbf{x})}\left[\frac{1}{H}\sum_{t=1}^{H} D_{\mathrm{KL}}\left(p_\theta(\cdot \mid \mathbf{x}, \mathbf{y}_{<t}) \,\|\, p_T(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})\right)\right]. \tag{8}$$

The loss is applied only to student-generated positions, meaning the prompt does not contribute to the loss. With this loss, the student recurrent states S_t that are generated during decoding are no longer outside the training distribution as the teacher supervises the student after its own deviations.

KL-Guided Rollout Horizon Doubling. Long rollouts are expensive and especially unstable early in OPD when the student already diverges from the teacher over short horizons. Instead, we adaptively increase rollout length throughout training. Training begins at $H = 512$, and doubles as follows,

$$\Delta D_{\mathrm{KL}}(p_\theta \,\|\, p_T) < \epsilon \text{ for } P \text{ updates} \quad \Longrightarrow \quad H \leftarrow \min(2H, H_{\max}), \tag{9}$$

where $H_{\max} = 32\mathrm{K}$ and ΔD_{KL} denotes the relative improvement of the reverse KL over its best value at the current rollout horizon. By increasing the rollout horizon only after KL plateaus, we train the student to remain stable over progressively longer sequences of recurrent state updates.

Sample Curriculum. We also progressively increase the difficulty of the OPD training distribution. Note that difficulty is controlled at the dataset level rather than estimated per sample. Once the rollout horizon reaches 1K, we introduce long-context retrieval examples spanning context lengths of 2K-32K tokens alongside reasoning and general instruction-following prompts. At later OPD checkpoints, we further replace the initial reasoning data with harder mathematical problems.

Prolonged OPD with WSD. Long-horizon capabilities continue to improve as we increase the amount of on-policy training. But, the required training budget is difficult to determine a priori. To enable effective checkpointing, we use a WSD learning rate schedule that lets us incrementally extend OPD while retaining annealed checkpoints at intermediate budgets. At each target token budget B_k, we save the full training state immediately before the decay phase. In this way, we can branch from this predecayed checkpoint to both evaluate the checkpoint and continue OPD training

$$\theta_k^{\mathrm{pre}} \begin{cases} \xrightarrow{\text{decay}} \theta_k^{\mathrm{eval}}, \\ \xrightarrow{\text{extend}} \theta_{k+1}^{\mathrm{pre}} \xrightarrow{\text{decay}} \theta_{k+1}^{\mathrm{eval}}, \end{cases} \qquad B_{k+1} > B_k. \tag{10}$$

This procedure lets us scale a single OPD trajectory to 2B OPD tokens while obtaining comparable checkpoints throughout. Beyond extending the Stage 3 training formula efficiently, it allows for direct measurement of long-context retrieval and reasoning recover as a function of OPD compute.

Unlike standard OPD settings that begin from an already long-context capable student, our post-Stage 2 student remains substantially weaker than the teacher on long contexts. Consequently, early student rollouts frequently depart from regions where the teacher and student behave similarly, and recovery requires sustained on-policy training as the student's trajectory distribution evolves. We use the original full-attention model as the teacher rather than a stronger model as our objective is to recover capabilities lost during architectural conversion rather than train capabilities that the original model did not possess. This combination of an initially weak student and the use of the original full-attention model, rather than a stronger teacher, makes prolonging the OPD stage necessary.

As the only layers with direct access to the full token history, the retained full attention layers must support long-range interaction in ways that the GDN layers with their compressed state cannot. While prior approaches often keep these retained attention layers frozen during distillation, we keep them trainable. Stage 2 uses two step sizes, before Stage 3 switches to the same flat learning rate for all layers so that the GDN and retained full-attention layers can train jointly.

4 RESULTS

4.1 EXPERIMENTAL SETUP

Models and Datasets. We apply OPAL to Qwen3-4B and MiMo-7B-RL-0530. Both contain 36 transformer layers, so our linearization pipeline results in 9 full attention layers and 27 GDN layers. Stages 1 and 2 use a pretraining-style mixture of FineWeb-Edu (Penedo et al., 2024), CodeParrot-Clean, and OpenWebMath (Paster et al., 2024). For OPD, we sample prompts from OpenThoughts-114K-Math (Guha et al., 2026), Dolly-15K (Conover et al., 2023), and synthetic long-context retrieval tasks. Subsequent checkpoints are also trained with DeepScaleR (Luo et al., 2025) and DAPO-Math-17K (Yu et al., 2025) in place of the OpenThoughts-114K-Math to encourage performance on harder tasks. The datasets are filtered by 13-gram overlap with our evaluation tasks.

We evaluate three aspects of model quality. For short-context retention, we use PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), ARC-Easy (Clark et al., 2018), ARC-Challenge, WinoGrande (Sakaguchi et al., 2021), and MMLU (Hendrycks et al., 2021), using zero-shot evaluation except for 5-shot MMLU. For long-context retrieval, we evaluate single needle, multi-key, and multi-query RULER-style tasks (Hsieh et al., 2024) at 4K, 8K, 16K, and 32K context lengths. Finally, we evaluate mathematical reasoning on GSM8K (Cobbe et al., 2021), GSM-Plus (Li et al., 2024a), GSM-Symbolic (Mirzadeh et al., 2025), MATH-500, AIME'24 (Zhang & Math-AI, 2024), and AIME'25 (Zhang & Math-AI, 2025). We report pass@1 for GSM8K, GSM-Plus, GSM-Symbolic, and MATH-500 and pass@1 and pass@8 for AIME, the most challenging reasoning task.

Training Parameters. Each student is trained on approximately 3B tokens: 100M for attention alignment, 600M for short-context distillation, 300M for 32K long-context distillation, and 2B student-generated tokens for OPD. Alignment and short-context KD use 4K sequences, while long-context KD uses 32K sequences. During OPD, rollouts begin at 512 tokens and iteratively increase in length until 32K. Complete optimization hyperparameters are provided in the Appendix.

Baselines. We compare against publicly released linearized models from RADLADS (Goldstein et al., 2025), Llamba (Bick et al., 2025), Zebra-Llama (Yang et al., 2025b), and HALO (Chen et al., 2026). We compare each student with the full-attention teacher from which it is distilled. Because these approaches differ substantially in teacher strength and training budget, we emphasize recovery relative to the full-attention teacher in addition to absolute performance. To isolate the effect of OPD, we also train off-policy controls using the same Qwen3-4B hybrid initialization and training schedule, but instead distill on fixed dataset sequences, similar to the Stage 2B portion.

4.2 MAIN RESULTS

In Tables 1 and 2, we demonstrate the performance of our OPAL-linearized models relative to prior linearization baselines for NIAH and math reasoning tasks. On NIAH, OPAL-Qwen3 matches its teacher with an average accuracy of 100.0%. The strongest baseline, Zebra-Llamba-8B only recovers 68% of its teacher's performance, 32% lower than both of our linearized models. These gains also extend to mathematical reasoning. OPAL-MiMo retains 93% of its teacher's average math reasoning performance and completely recovers AIME'24 pass@8. Its absolute math reasoning av-

Table 1: Needle-in-a-Haystack retrieval accuracy (%). *Teacher Model*; Student Model.

	Single Needle				Multi-Key				Multi-Query				
Model	**4K**	**8K**	**16K**	**32K**	**4K**	**8K**	**16K**	**32K**	**4K**	**8K**	**16K**	**32K**	**Avg.**
Qwen2.5-7B-Instruct	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
QRWKV6-7B-Instruct	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1 (0.1%)
Llama-3.2-1B-Instruct	100.0	100.0	100.0	100.0	94.0	93.0	71.0	67.0	98.5	94.2	93.2	96.5	92.3
Llamba-1B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0 (0.0%)
Llama-3.2-3B-Instruct	100.0	100.0	100.0	100.0	98.0	95.0	94.0	87.0	100.0	99.2	87.5	68.5	94.1
Llamba-3B	2.0	0.0	0.0	0.0	3.0	3.0	0.0	0.0	0.8	0.0	0.0	0.0	0.7 (0.8%)
Zebra-Llama-3B	85.0	61.0	52.0	47.0	55.0	39.0	28.0	10.0	64.2	24.2	23.0	3.5	41.0 (43.6%)
Llama-3.1-8B-Instruct	100.0	100.0	100.0	100.0	100.0	100.0	99.0	99.0	100.0	100.0	100.0	100.0	99.8
Zebra-Llama-8B	**100.0**	98.0	96.0	74.0	73.0	40.0	40.0	27.0	86.8	70.0	66.5	43.0	67.9 (68.0%)
Qwen3-1.7B	100.0	100.0	100.0	100.0	99.0	100.0	99.0	99.0	100.0	100.0	100.0	100.0	99.8
HypeNet-2B	73.0	76.0	66.0	73.0	28.0	22.0	24.0	18.0	27.3	19.0	18.5	15.5	38.4 (38.5%)
Qwen3-4B	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.0	100.0	100.0	100.0	100.0	99.9
HypeNet-5B	88.0	90.0	90.0	96.0	34.0	37.0	33.0	28.0	49.5	55.8	49.2	45.0	58.0 (58.0%)
OPAL-Qwen3 (ours)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**99.6**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0** (100.0%)
MiMo-7B-RL-0530	100.0	100.0	100.0	100.0	99.6	99.0	98.8	92.8	99.9	99.7	98.3	76.2	97.0
OPAL-MiMo (ours)	**100.0**	**100.0**	**100.0**	**100.0**	96.8	96.8	94.8	83.8	**100.0**	**100.0**	**100.0**	**100.0**	97.7 (100.7%)

Table 2: Mathematical reasoning performance (%). *Teacher Model*; Student Model.

					AIME'24		AIME'25		
Model	**GSM8K**	**GSM-Plus**	**GSM-Sym**	**MATH-500**	**pass@1**	**pass@8**	**pass@1**	**pass@8**	**Avg.**
Qwen2.5-7B-Instruct	91.9	72.7	83.4	76.4	10.4	26.7	7.5	23.3	49.0
QRWKV6-7B-Instruct	27.4	16.8	9.3	24.0	0.4	3.3	0.4	3.3	10.6 (21.6%)
Llama-3.2-1B-Instruct	38.5	24.9	26.2	25.8	0.8	6.7	0.0	0.0	15.4
Llamba-1B	28.1	12.9	13.0	8.6	0.0	0.0	0.0	0.0	7.8 (51.0%)
Llama-3.2-3B-Instruct	71.1	58.3	59.5	39.0	2.5	13.3	0.0	0.0	30.5
Llamba-3B	50.7	27.5	24.8	7.8	0.4	3.3	0.0	0.0	14.3 (47.0%)
Zebra-Llama-3B	62.7	38.8	36.7	20.0	0.4	3.3	0.0	0.0	20.2 (66.4%)
Llama-3.1-8B-Instruct	86.1	67.0	73.2	48.6	2.9	10.0	0.0	0.0	36.0
Zebra-Llama-8B	61.3	40.3	35.3	24.4	0.8	6.7	0.4	3.3	21.6 (59.9%)
Qwen3-1.7B	83.5	63.3	65.0	74.6	13.8	30.0	10.4	23.3	45.5
HypeNet-2B	1.1	3.8	1.2	4.4	0.0	0.0	0.8	3.3	1.8 (4.0%)
Qwen3-4B	92.3	72.8	83.2	85.4	25.8	43.3	22.1	43.3	58.5
HypeNet-5B	14.9	15.8	8.7	11.8	0.0	0.0	0.0	0.0	6.4 (10.9%)
Qwen3-4B	92.3	72.8	83.2	96.4	73.8	86.7	63.7	80.0	81.1
OPAL-Qwen3 (ours)	83.2	**63.8**	**65.9**	92.0	52.5	80.0	40.4	63.3	67.6 (83.4%)
MiMo-7B-RL-0530	80.2	61.6	71.6	97.0	73.3	83.3	67.9	83.3	77.3
OPAL-MiMo (ours)	**84.8**	63.5	65.7	**93.8**	**65.4**	**83.3**	**47.5**	**73.3**	**72.2** (93.4%)

erage is also 50.6% higher than the next strongest baseline. As HypeNet-2B and HypeNet-5B are not trained with thinking mode enabled, they must be compared to the performance of their non-thinking teachers. However, HypeNet-5B, which has the same teacher model as OPAL-Qwen3, lags behind OPAL-Qwen3's average math reasoning performance by 61.2%.

This improved long-context performance comes with a tradeoff on commonsense reasoning benchmarks, whose full results we defer to the Appendix for space. OPAL-Qwen3 retains 87% of its teacher's performance on the commonsense suite, while OPAL-MiMo retains 94% with absolute scores remaining within the range achieved by prior linearized models. Although we acknowledge the tradeoff in commonsense performance, these results reflect a shift toward preserving the long-context capabilities that motivate hybrid attention in the first place. OPAL achieves these improvements with a comparatively small training budget and without separate post-training phases beyond distillation. RADLADS and HALO both use SFT for post-training, whereas Zebra-Llama incorporates a short DPO stage after SFT. Zebra-Llama-3B and Zebra-Llama-8B use 9B and 11B training tokens, respectively. In contrast, OPAL-Qwen3 and OPAL-MiMo only use 3B training tokens.

Reasoning Scales with OPD Compute. Figure 3 shows how mathematical reasoning performance evolves with additional OPD training. Performance improves substantially as the number of OPD tokens increases across both models. On MATH-500, most of the improvement occurs early, after which performance saturates near the teacher. In contrast, AIME performance continues to improve throughout training with the strongest checkpoints occurring near the end of our 2B token training budget. This suggests that OPD compute is especially valuable for more difficult reasoning tasks and, since AIME has not clearly saturated, that further scaling may yield additional gains.

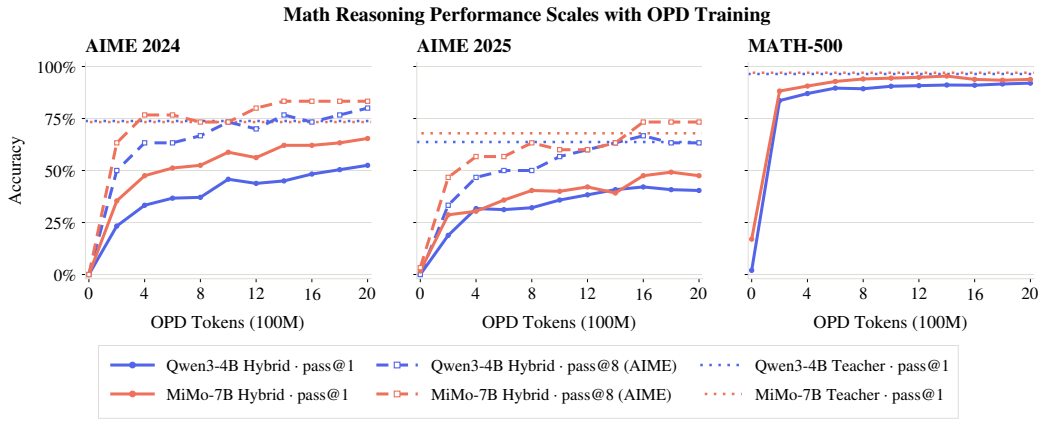


Figure 3: Math reasoning performance as a function of OPD training tokens. MATH-500 improves rapidly and saturates early, while AIME continues to benefit from additional OPD compute.

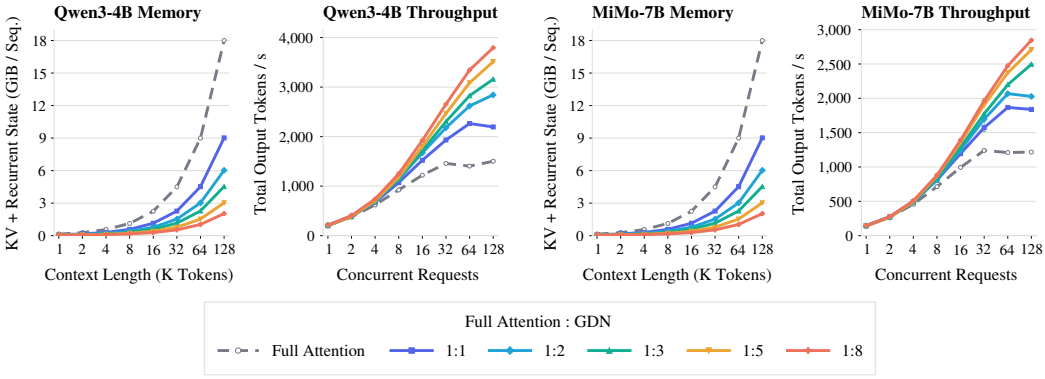


Figure 4: Memory and throughput improvements of GDN hybrids over the full attention teacher.

Comparison to Off-Policy Controls. To isolate the benefit of training on student-generated trajectories, we compare OPD against matched off-policy baselines. In these controls, the post-Stage 2 hybrid student is trained on the Stage 3 data mixture but using off-policy KD. The traces used for training are either the dataset ground-truth text, as in Stage 2, or teacher traces generated by the full attention model. Each experiment uses Qwen3-4B as the model backbone. The off-policy baselines significantly underperform relative to OPD. At 400M tokens, OPD improves over the extended Stage 2 recipe by 16.2% on AIME'24 and 15.9% on AIME'25. Training on teacher-generated traces marginally improves performance, but this regime still lags behind OPD by up to 15.8%. The results support the hypothesis that OPD is necessary for correcting errors induced by the recurrent state in hybrid models over long contexts.

Table 3: OPD versus off-policy controls at matched training budgets. Parenthesis show source of training traces.

	MATH-500		AIME'24		AIME'25	
Training	**200M**	**400M**	**200M**	**400M**	**200M**	**400M**
OPD (Student)	**83.6**	**87.0**	**23.3**	**33.3**	**18.8**	**31.7**
Off-Policy (Dataset)	77.3	78.1	12.5	17.1	14.6	15.8
Off-Policy (Teacher)	75.5	80.0	12.9	17.5	13.8	16.7

4.3 Additional Experiments

Efficiency Gains. Figure 4 evaluates the inference efficiency gains from attention linearization. We serve both the full attention and hybrid models using vLLM. At the 1:3 full attention to GDN ratio used by OPAL, cache and recurrent-state memory is approximately 4× lower than full attention at 128K context, where KV cache memory dominates overall memory. At the highest concurrency, throughput is approximately 2× higher for both Qwen and MiMo hybrids. Thus, OPAL preserves the efficiency benefits that motivate hybrid attention architectures in the first place.

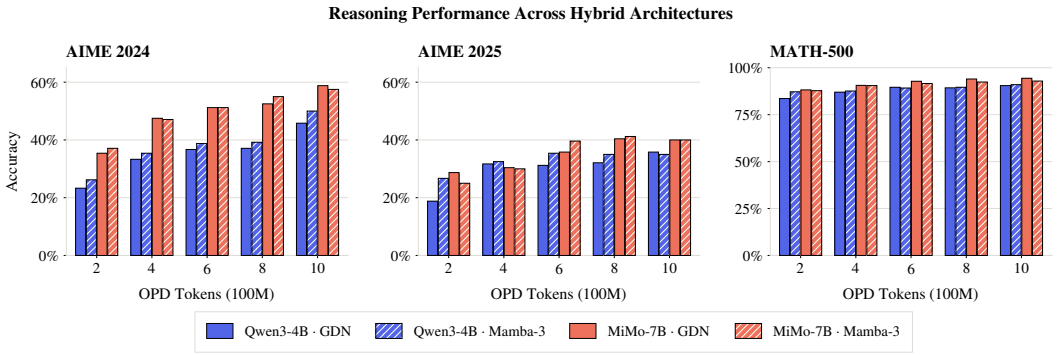


Figure 5: Comparison of how reasoning scales for various hybrid architectures.

Table 4: Comparison to non-distillation baselines on NIAH (32K) and math reasoning (%).

	NIAH, 32K Context			Math Reasoning		
Model / Conversion	**Single Needle**	**Multi-Key**	**Multi-Query**	**MATH-500**	**AIME'24**	**AIME'25**
Qwen3-4B	100.0	99.8	100.0	96.4	73.8	63.7
↪ SWA(256, 4)	0.0	1.0	0.2	17.4	0.0	0.0
↪ GDN (ours, 2B OPD)	**100.0**	**99.6**	**100.0**	**92.0**	**52.5**	**35.0**
MiMo-7B-RL-0530	100.0	92.8	76.2	97.0	73.3	67.9
↪ SWA(256, 4)	0.0	1.0	0.3	51.0	1.7	1.7
↪ TransMLA (3B Recovery)	39.0	2.0	1.5	27.0	0.4	0.0
↪ GDN (ours, 2B OPD)	**100.0**	**83.8**	**100.0**	**93.8**	**59.6**	**47.1**

Alternative Linear Attention Architectures. We apply OPAL to Qwen3-4B and MiMo-7B-RL-0530 using Mamba-3 (Lahoti et al., 2026). Figure 5 shows that Mamba-3 hybrids achieve reasoning performance competitive with GDN at matched OPD budgets. At 1B OPD tokens, Qwen3 with Mamba-3 reaches 91.0%, 50.0%, and 35.0% on MATH-500, AIME'24, and AIME'25, respectively. At 800M tokens, MiMo-7B-RL-0530 with Mamba-3 reaches 92.4%, 55.0%, and 41.2%. Thus, the OPAL procedure is not GDN-specific and can applied to models using various efficient attention architectures while maintaining reasoning and long-context capabilities.

Comparison to Non-Distillation Baselines. We compare OPAL against two non-distillation attention conversions (Table 4). First, we convert both Qwen3-4B and MiMo-7B-RL-0530 to hybrid sliding window attention (SWA) using a 256-token window and four sinks (Beltagy et al., 2020; Jolicoeur-Martineau et al., 2026). We retain the same 1:3 full attention to converted layer ratio as our GDN hybrids to approximately match long-context memory budgets. For MiMo-7B-RL-0530, we convert the model to multi-head latent attention (MLA) (DeepSeek-AI, 2024) by following the TransMLA (Meng et al., 2025) conversion recipe with 3B recovery SFT tokens. The converted SWA and TransMLA models incur substantial losses in reasoning and long-context retrieval. Our OPAL-Qwen3 and OPAL-MiMo hybrids reach 92.0% and 93.8% on MATH-500 versus 17.4% and 51.0% for SWA and 27.0% for MiMo-7B-RL-0530 with TransMLA. On AIME and 32K token context NIAH, our OPAL hybrids outperform both conversion baselines which completely collapse.

5 CONCLUSION

We present OPAL, a method for converting fully trained, full-attention transformers to hybrid models while maintaining their long-context retrieval and reasoning abilities. Traditional off-policy distillation baselines remain performant on likelihood tasks, but suffer performance degradation on more involved, long-horizon benchmarks. OPAL addresses this failure mode with a prolonged, continual-learning-style OPD stage, providing a training signal for the hybrid student to learn from the states it encounters during generation. This is especially relevant for hybrid students whose compressed recurrent state can accumulate errors over long contexts. When applied to Qwen3-4B and MiMo-7B-RL-0530, OPAL fully recovers long-context retrieval performance and closes the gap on mathematical reasoning while retaining only a quarter of the full-attention layers. Broadly, our findings position on-policy training as an important ingredient for post-hoc architectural conversion, and we aim to apply a similar training recipe to other architectural conversions to show that capabilities do not have to be tied to the architecture in which they were originally learned.

AI USE STATEMENT

We used generative AI tools to polish and compress manuscript text to meet the page limit and to search for related work that may have been missed in the initial draft. We also used AI agents to assist with writing code and launching experiments on GPU/compute clusters. Generative AI was not used for the ideation or formulation of the research problem. The authors reviewed the AI-assisted text, references, code, and experimental outputs and take responsibility for the final content and reported results.

ETHICS STATEMENT

This work studies how to convert existing full-attention language models into more memory-efficient hybrid models. We do not identify a new application-specific ethical risk from the conversion method beyond what is already posed by existing models/methods. The authors have read the ICLR code of ethics and believe that the paper complies with it.

REPRODUCIBILITY STATEMENT

We provide the training hyperparameters in the Appendix and the anonymized training code in a ZIP file submitted as supplementary material. We plan to release all assets including the code and model checkpoints upon publication.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Y Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International conference on learning representations*, volume 2024, pp. 15664–15730, 2024.

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise LoRA. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=WwpYSOkkCt.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

Aviv Bick, Kevin Y. Li, Eric P. Xing, J. Zico Kolter, and Albert Gu. Transformers to SSMs: Distilling quadratic knowledge to subquadratic models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0999. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3848fef259495bfd04d60cdc5c1b4db7-Abstract-Conference.html.

Aviv Bick, Tobias Katsch, Nimit Sharad Sohoni, Arjun D Desai, and Albert Gu. Llamba: Scaling distilled recurrent models for efficient language processing. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025. URL https://openreview.net/forum?id=uciWntM6iv.

Aviv Bick, Eric P. Xing, and Albert Gu. Retrieval-aware distillation for transformer-SSM hybrids. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=8gsYox51jQ.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5209–5235. PMLR, 2024. URL https://proceedings.mlr.press/v235/cai24b.html.

Aditya Chattopadhyay, Elvis Nunez, Prannay Kaul, Benjamin Bowman, Evan Becker, Luca Zancato, David Thomas, Wei Xia, and Stefano Soatto. Priming: Hybrid state space models from pre-trained transformers, 2026. URL https://arxiv.org/abs/2605.08301.

Yingfa Chen, Zhen Leng Thai, Zihan Zhou, Zhu Zhang, Xingyu Shen, Shuo Wang, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Hybrid linear attention done right: Efficient distillation and effective architectures for extremely long contexts, 2026. URL https://arxiv.org/abs/2601.22156.

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, 2018. URL https://arxiv.org/abs/1803.05457.

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, 2021. URL https://arxiv.org/abs/2110.14168.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Mahabaleshwarkar, Shih-Yang Liu, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, et al. Hymba: A hybrid-head architecture for small language models. In *International Conference on Learning Representations*, volume 2025, pp. 97729–97755, 2025.

Daniel Goldstein, Eric Alcaide, Janna Lu, and Eugene Cheah. RADLADS: Rapid attention distillation to linear attention decoders at scale. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=38GehGepDd.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Etash Kumar Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Rea Sprague, Ashima Suvarna, Benjamin Feuer, Leon Liangyu Chen, Zaid Khan, and Eric Frankel et. al. Openthoughts: Data recipes for reasoning models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7xjoTuaNmN.

Daya et al. Guo. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=3X2L2TFr0f.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and eran malach. Repeat after me: Transformers are better than state space models at copying. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=duRRoGeoQT.

Alexia Jolicoeur-Martineau, Rhea Sanjay Sukthanker, Pashmina Cameron, and Emy Gervais. Sliding-window beats linear attention, 2026. URL https://arxiv.org/abs/2608.28444.

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. Finetuning pretrained transformers into RNNs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10630–10643. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.830. URL https://aclanthology.org/2021.emnlp-main.830/.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation, 2016. URL https://arxiv.org/abs/1606.07947.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=T5nUQDrM4u.

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*, pp. 611–626, 2023.

Aakash Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, J Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HwCvaJOiCj.

Disen Lan, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng. Liger: Linearizing large language models to gated recurrent structures. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 32452–32466. PMLR, 2025. URL https://proceedings.mlr.press/v267/lan25b.html.

Disen Lan, Jianbin Zheng, Yuxi Ren, Xin Xia, Xuanda Wang, Xuefeng Xiao, Xipeng Qiu, and Yu Cheng. Morphing into hybrid attention models, 2026. URL https://arxiv.org/abs/2606.30562.

Barak Lenz, Opher Lieber, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, Daniel Gissin, Daniel Jannai, Dor Muhlgay, Dor Zimberg, Edden M. Gerber, Elad Dolev, Eran Krakovsky, Erez Safahi, Erez Schwartz, and Gal Cohen et. al. Jamba: Hybrid transformer-mamba language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JFPaD7lpBD.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers, 2024a.

Yanhong Li, Songlin Yang, Shawn Tan, Mayank Mishra, Rameswar Panda, Jiawei Zhou, and Yoon Kim. Distilling to hybrid attention models via KL-guided layer selection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RzbsHcFqIf.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948. PMLR, 2024b. URL https://proceedings.mlr.press/v235/li24bt.html.

LLM-Core Xiaomi. MiMo-V2-Flash technical report, 2026. URL https://arxiv.org/abs/2601.02780.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-teacher on-policy distillation for capability integration in LLM post-training, 2026. URL https://arxiv.org/abs/2606.30406.

Fanxu Meng, Pingzhi Tang, Zengwei Yao, Xing Sun, and Muhan Zhang. TransMLA: Migrating GQA models to MLA with full DeepSeek compatibility and speedup. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2741. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/75d13a472f570755af2a4ae4ac3d6724-Abstract-Conference.html.

Jean Mercat, Igor Vasiljevic, Sedrick Keh, Kushal Arora, Achal Dave, Adrien Gaidon, and Thomas Kollar. Linearizing large language models, 2024. URL https://arxiv.org/abs/2405.06640.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AjXkRZIvjB.

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 *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1299. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4822991365c962105b1b95b1107d30e5-Abstract-Conference.html.

NVIDIA. Nemotron 3 Ultra: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning, 2026. URL https://arxiv.org/abs/2606.15007.

NVIDIA, :, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, Aleksandr Shaposhnikov, Alex Kondratenko, Alexander Bukharin, Alexandre Milesi, Ali Taghibakhshi, Alisa Liu, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amir Klein, Amit Zuker, Amnon Geifman, Amy Shen, Anahita Bhiwandiwalla, Andrew Tao, Ann Guan, Anubhav Mandarwal, Arham Mehta, and Ashwath Aithal et.al. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning, 2025. URL https://arxiv.org/abs/2512.20848.

OpenAI. Openai o1 system card, 2026. URL https://arxiv.org/abs/2412.16720.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jKHmjlpViu.

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 *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Liliang Ren, Yang Liu, Yadong Lu, yelong shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bIlnpVM4bc.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

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.

Mark Shapiro. Retrofitting recurrent depth into a pretrained language model: Installation, extrapolation, transfer, and retention at two parameter budgets, 2026. URL https://arxiv.org/abs/2608.11233.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2023. URL https://arxiv.org/abs/2307.08621.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, and Dazhi Cheng et. al. Kimi k3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Qwen Team. Qwen3.5-omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. The mamba in the llama: Distilling and accelerating hybrid models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1996. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/723933067ad315269b620bc0d2c05cba-Abstract-Conference.html.

Kaiyue Wen, Zhiyuan Li, Jason S. Wang, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=m51BgoqvbP.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations*, volume 2024, pp. 5385–5409, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Mingyu Yang, Mehdi Rezagholizadeh, Guihong Li, Vikram Appia, and Emad Barsoum. Zebra-llama: Towards extremely efficient hybrid models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=l42UGsdrNn.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ia5XvxFUJT.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=r8H7xhYPwz.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4g02l2N2Nx.

Michael Zhang, Simran Arora, Rahul Chalamala, Benjamin Frederick Spector, Alan Wu, Krithik Ramesh, Aaryan Singhal, and Christopher Re. LoLCATs: On low-rank linearizing of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8VtGeyJyx9.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Zhongzhu Zhou, Qingyang Wu, Junxiong Wang, Mayank Mishra, Shuaiwen Leon Song, Ben Athiwaratkun, and Chenfeng Xu. Taylor-calibrate: Principled initialization for hybrid linear attention distillation, 2026. URL https://arxiv.org/abs/2606.16429.

APPENDIX

A RELATED WORKS

Linear Attention and Hybrid Models. Linear attention and recurrent sequence models replace softmax attention with a fixed-size state rather than retaining the full history of key-value pairs, enabling subquadratic sequence processing (Katharopoulos et al., 2020; Gu & Dao, 2024; Dao & Gu, 2024; Yang et al., 2025c). However, pure fixed-state models exhibit limitations in recall and retrieval (Arora et al., 2024; Jelassi et al., 2024). While these architectures are competitive at smaller scales, many large-scale efficient language models still retain some form of full attention, motivating hybrid architectures (Lenz et al., 2025; Team et al., 2026; Qwen Team, 2026b). Hybrid models introduce interesting design choices, including the ratio and placement of full and linear attention layers (Lenz et al., 2025), the choice of recurrent or linear attention mixer, and whether the two mechanisms are interleaved across layers or combined within the same layer (Dong et al., 2025). A complementary design is explored by Samba (Ren et al., 2025), which combines Mamba with sliding window attention rather than relying on a small number of global full-attention layers.

Architecture Conversion after Pretraining. Architectural modification after pretraining is an emerging approach to improving model efficiency without paying the cost of retraining from scratch. At the lightest end of this spectrum, new modules can be added while leaving the pretrained backbone largely or entirely unchanged, such as auxiliary decoding heads for speculative generation (Cai et al., 2024; Li et al., 2024b) or gated cross attention layers that adapt a frozen language model to multimodal inputs (Alayrac et al., 2022). More substantial conversions modify the pretrained backbone itself. Dense models can be upcycled into sparse mixture-of-experts (MoE) models by converting dense FFNs into experts and continuing pretraining (Komatsuzaki et al., 2023), while autoregressive language models can be adapted into diffusion models by relaxing causal attention to bidirectional attention (Gong et al., 2025). The attention mechanism itself can be replaced after pretraining, including MHA-to-GQA and GQA-to-MLA conversion at a small fraction of the original pretraining cost (Ainslie et al., 2023; Meng et al., 2025). Pretrained layers can similarly be tied and reused recurrently to trade parameters for additional computational depth (Bae et al., 2025; Shapiro, 2026), while structured pruning followed by continued training or off-policy distillation can produce smaller models while recovering much of the original performance (Xia et al., 2024; Muralidharan et al., 2024). These approaches are evaluated on different benchmarks and sometimes do not recover long-context performance despite involving less disruptive architectural changes. Methods targeting capability recovery have generally relied on significant continued pretraining.

Hybridizing Full-Attention Transformers. Existing methods for converting pretrained full-attention transformer layers into linear attention layers typically use some combination of layerwise alignment (Stage 1), end-to-end off-policy distillation (Stage 2), and continual pretraining on additional data. Lightweight approaches primarily align each converted layer with the attention layer it replaces, learning feature maps or recurrent parameters while largely preserving the pretrained weights (Kasai et al., 2021; Zhang et al., 2024; 2025; Lan et al., 2025). A second family supplements such initialization with end-to-end off-policy distillation, matching the converted student's predictions to a frozen full-attention teacher on fixed sequences (Bick et al., 2024; Wang et al., 2024; Bick et al., 2025; Goldstein et al., 2025; Yang et al., 2025b; Chen et al., 2026; Li et al., 2026; Bick et al., 2026; Lan et al., 2026). Other approaches instead recover the converted model through substantial additional next-token training (Mercat et al., 2024; Chattopadhyay et al., 2026). While all these methods typically recover short-context quality effectively, long-context retrieval often lags behind the source transformer while long-context reasoning/generation typically collapses. Priming (Chattopadhyay et al., 2026) is a notable exception, demonstrating strong long-context reasoning with a 1:1 full attention:GDN/SSM-attention hybrid, but requires multi-stage continual pretraining at context lengths up to 128K, supervised fine-tuning on chain-of-thought traces, and instruction alignment while using 150B tokens, approaching the Chinchilla compute-optimal (Hoffmann et al., 2022) pretraining budget (160B) for the 8B parameter model considered.

On-Policy Distillation and Long-Horizon Reasoning. Reasoning models often generate long trajectories, making the distribution of model-generated prefixes increasingly important as generation proceeds. Recent progress in reasoning has been driven largely by reinforcement learning with verifiable rewards (RLVR), with GRPO and its variants providing a scalable means of optimizing directly on model-generated trajectories (Shao et al., 2024; Guo, 2025; Yu et al., 2025). On-policy

distillation has emerged as a complementary approach, replacing sparse outcome rewards with dense token-level supervision from a stronger teacher on the student's own generations (Agarwal et al., 2024; Lu & Lab, 2025). This has been particularly effective for transferring reasoning capabilities from larger to smaller models (Yang et al., 2025a). More recently, multi-teacher on-policy distillation has been used to integrate independently trained reasoning capabilities in frontier-scale post-training (Ma et al., 2026; LLM-Core Xiaomi, 2026). Notably, Nemotron 3 Ultra, a 550B-parameter MoE hybrid Mamba hybrid model, combines SFT, RL, and multi-teacher on-policy distillation in its post-training pipeline (NVIDIA, 2026), demonstrating that on-policy distillation is also applicable to large hybrid reasoning models.

B Experimental Details

B.1 Models and Hybrid Construction

Table 5: Teacher models and the converted hybrids. Both retain the same nine full attention layers and use Gated DeltaNet (GDN) as the linear attention layer.

	Qwen3-4B	MiMo-7B-RL-0530
Teacher	`Qwen/Qwen3-4B`	`XiaomiMiMo/MiMo-7B-RL-0530`
Teacher Parameters	4.02B	7.62B (MTP head dropped)
Layers	36, GQA 32 Q/8 KV, $d_h = 128$	36, GQA 32 Q/8 KV, $d_h = 128$
Context Length	32,768	65,536
Retained Full Attention Layers	9 (Layers $3, 7, \ldots, 35$)	
Converted Layers	27 (Layers 0, 1, 2, 4, 5, 6, …, 34)	
GDN Initialization	Q/K/V/O inherited from full attention teachers. Others set to default values	
Hybrid Student Parameters	4.55B (1.14B in GDN Layers)	8.31B (1.82B in GDN Layers)

B.2 Training Procedure and Hyperparameters

All runs were conducted using 3–4 NVIDIA H100 or H200 GPUs, depending on availability.

Table 6: Optimization hyperparameters by stage. All stages use AdamW (β_1=0.9, β_2=0.95, weight decay 0), gradient clipping at norm 1.0. "Inherited" denotes all non-GDN weights.

	Stage 1		Stage 2A		Stage 2B		Stage 3	
	Qwen	MiMo	Qwen	MiMo	Qwen	MiMo	Qwen	MiMo
Trainable	GDN	GDN	All	All	All	All	All	All
Tokens	100M	100M	600M	600M	294M	294M	2B	2B
Sequence length	4K	4K	4K	4K	32K	32K	32K	32K
Tokens per batch	32K	16K	2.1M	2.1M	2.1M	2.1M	131K	131K
Optimizer Steps	3.1K	6.1K	287	287	140	140	12K	12K
Peak LR, GDN	3×10^{-3}	1×10^{-3}	2×10^{-4}	5×10^{-5}	1×10^{-4}	2.5×10^{-5}	2×10^{-5}	1×10^{-5}
Peak LR, Inherited	–	–	2×10^{-5}	5×10^{-6}	1×10^{-5}	2.5×10^{-6}	2×10^{-5}	1×10^{-5}
Warmup Tokens	3M	3M	18M	18M	8.8M	8.8M	20M	20M
LR Schedule	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine	WSD	WSD

Table 7: Stage-3 on-policy distillation settings. Rollouts are sampled with vLLM at T=1.0, top-p=1.0 under the thinking-mode chat template.

	Qwen3-4B	MiMo-7B-RL-0530
Generated Tokens per Step	131,072	131,072
Prompt + Rollout Limit	40,960	65,536
Initial Horizon H	512	512
Horizon Doubling Rule	KL Plateau: rel. gain <2% for 30 steps	
Maximum Horizon	32K	32K
Sampler Batch (Prompts)	64; 32 at H=32k	64; 32 at H=32k
Peak LR	2×10^{-5}	1×10^{-5}
Warmup	150 Steps (20M Tokens)	150 Steps (20M Tokens)

B.3 OPD Scaling Ladder

Table 8: OPD scaling ladder configuration. Each rung continues the previous one from its pre-decay snapshot at the constant learning rate, then decays linearly to the final LR over the decay window.

Rung	Max Output Horizon	Decay Start	Decay Tokens
Qwen3-4B			
0 → 200M	512 → 16K	161M	39M
200M → 400M	16K	340M	60M
400M → 500M	16K	430M	70M
500M → 600M	16K	520M	80M
600M → 800M	16K → 32K	710M	90M
800M → 900M	32K	800M	100M
900M → 1.0B	32K	890M	110M
1.0B → 1.2B	32K	1.09B	110M
1.2B → 1.4B	32K	1.29B	110M
1.4B → 1.6B	32K	1.49B	110M
1.6B → 1.8B	32K	1.69B	110M
1.8B → 2.0B	32K	1.89B	110M
MiMo-7B-RL-0530			
0 → 200M	512 → 16K	161M	39M
200M → 400M	16K	340M	63M
400M → 600M	16K	537M	63M
600M → 800M	16K → 32K	737M	63M
800M → 1.0B	32K	937M	63M
1.0B → 1.2B	32K	1.14B	63M
1.2B → 1.4B	32K	1.34B	63M
1.4B → 1.6B	32K	1.52B	79M
1.6B → 1.8B	32k	1.72B	79M
1.8B → 2.0B	32k	1.92B	79M

C Additional Experiments

C.1 Commonsense Reasoning Performance

Table 9: Commonsense reasoning performance (%). *Teacher Model*; Student Model.

Model	PIQA	HellaSwag	ARC-E	ARC-C	Winogrande	MMLU	Avg.
Qwen2.5-7B-Instruct	80.2	80.5	80.9	55.0	70.5	74.3	73.6
QRWKV6-7B-Instruct	**79.9**	78.9	**79.3**	55.9	71.4	**64.2**	**71.6** (97.3%)
Llama-3.2-1B-Instruct	74.9	61.6	63.9	37.6	61.4	46.0	57.6
Llamba-1B	73.7	61.8	65.2	37.5	61.9	31.5	55.3 (96.0%)
Llama-3.2-3B-Instruct	76.8	71.6	71.0	46.4	68.7	60.7	65.9
Llamba-3B	78.0	74.0	73.6	46.2	**71.7**	50.1	65.6 (99.6%)
Zebra-Llama-3B	77.3	72.7	75.8	52.3	65.7	51.7	65.9 (100.1%)
Llama-3.1-8B-Instruct	81.4	79.5	79.9	55.6	73.6	68.4	73.1
Zebra-Llama-8B	**79.9**	**79.0**	77.5	**57.6**	71.3	59.1	70.7 (96.8%)
Qwen3-1.7B	72.0	60.4	69.5	43.0	61.6	60.2	61.1
HypeNet-2B	72.0	57.2	67.3	42.7	61.7	39.5	56.7 (92.8%)
Qwen3-4B	74.9	68.5	78.5	53.8	65.8	70.1	68.6
HypeNet-5B	76.3	66.1	75.9	50.6	67.2	52.4	64.8 (94.4%)
OPAL-Qwen3 (ours)	72.3	58.5	65.0	41.6	63.0	57.2	59.6 (86.9%)
MiMo-7B-RL-0530	74.5	64.9	68.1	45.0	61.6	57.2	61.9
OPAL-MiMo (ours)	72.5	59.9	62.7	39.5	58.8	54.6	58.0 (93.7%)

C.2 WSD Training Curves

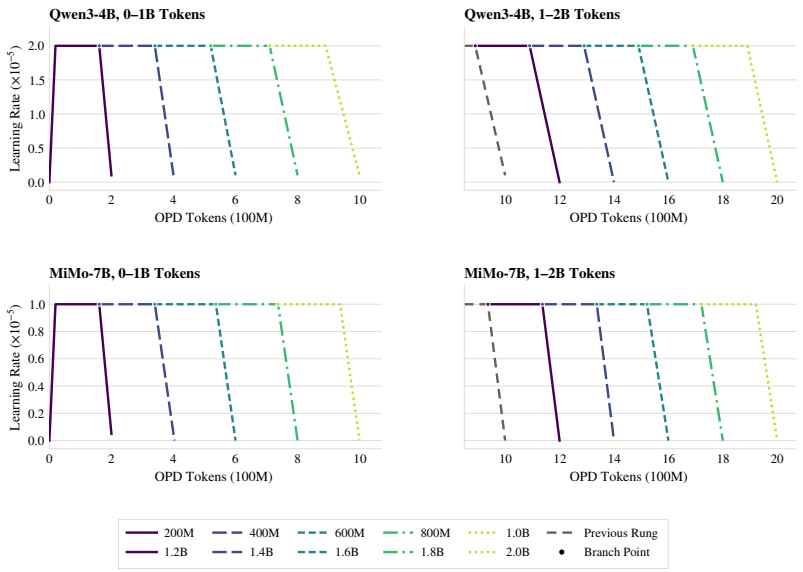


Figure 6: Training horizons for each rung of the OPD scaling ladder. Decay horizons gradually become longer at subsequent rungs before plateauing in the last few rungs.

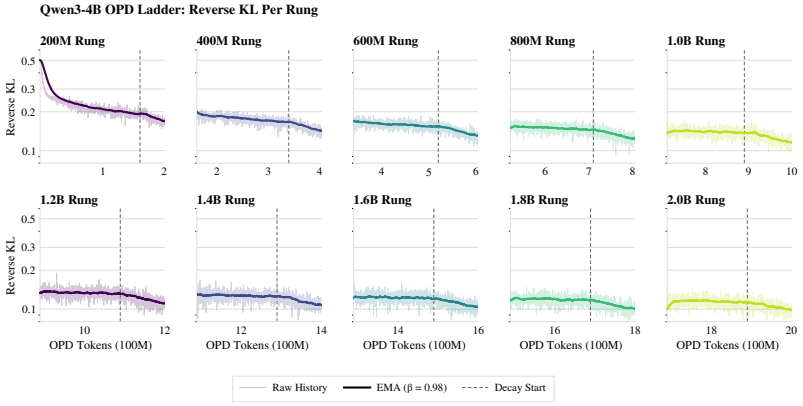


Figure 7: Training curves for Qwen3-4B at each rung using WSD schedule.

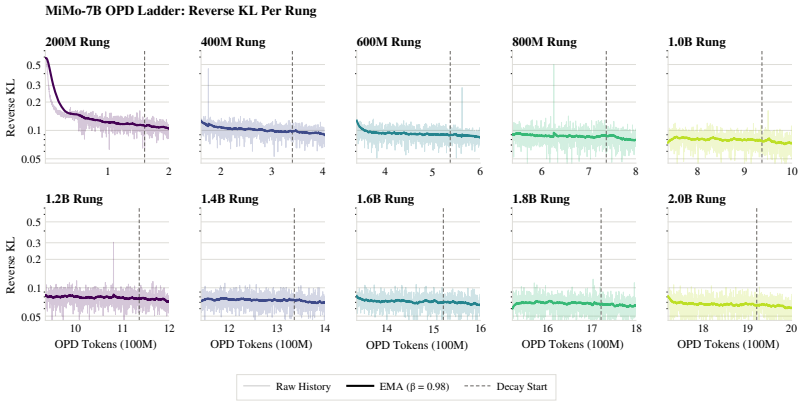


Figure 8: Training curves for MiMo-7B-RL-0530 at each rung using WSD schedule.

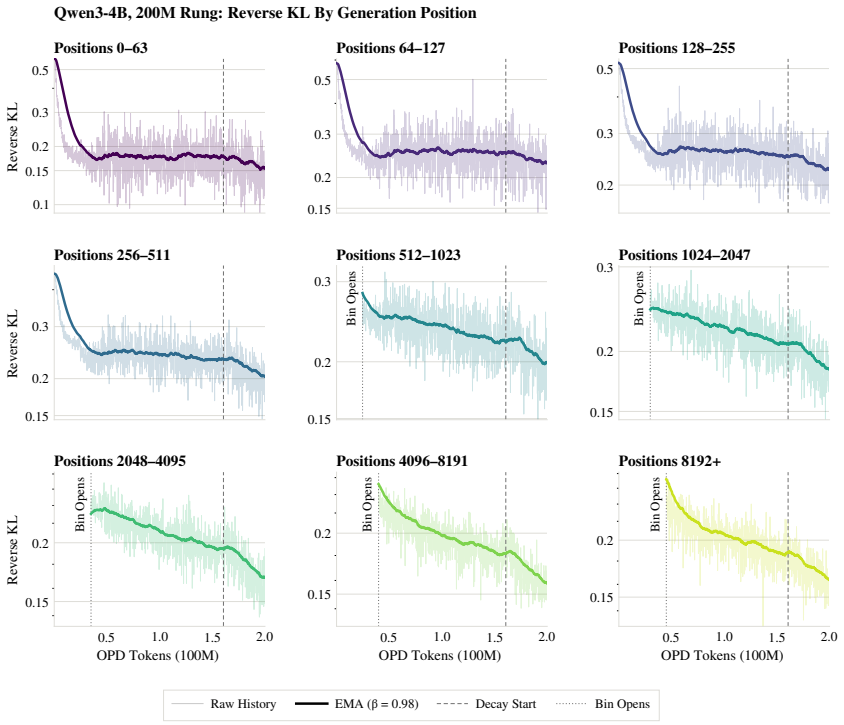


Figure 9: Training curves for position-wise reverse-KL at the first rung of Qwen3-4B training.

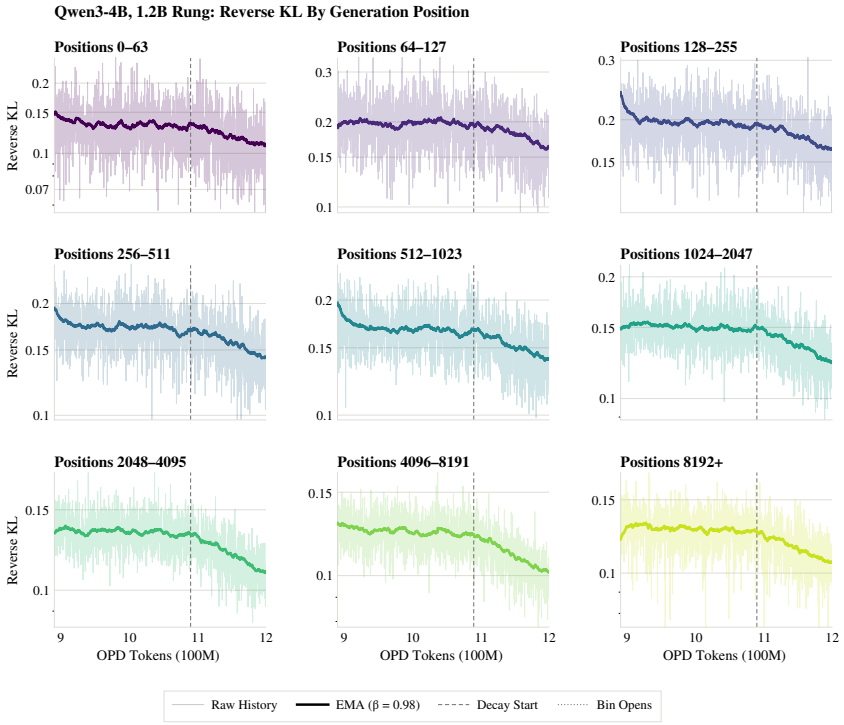


Figure 10: Training curves for position-wise reverse-KL at the run 1B-1.2B OPD tokens of Qwen3-4B training.

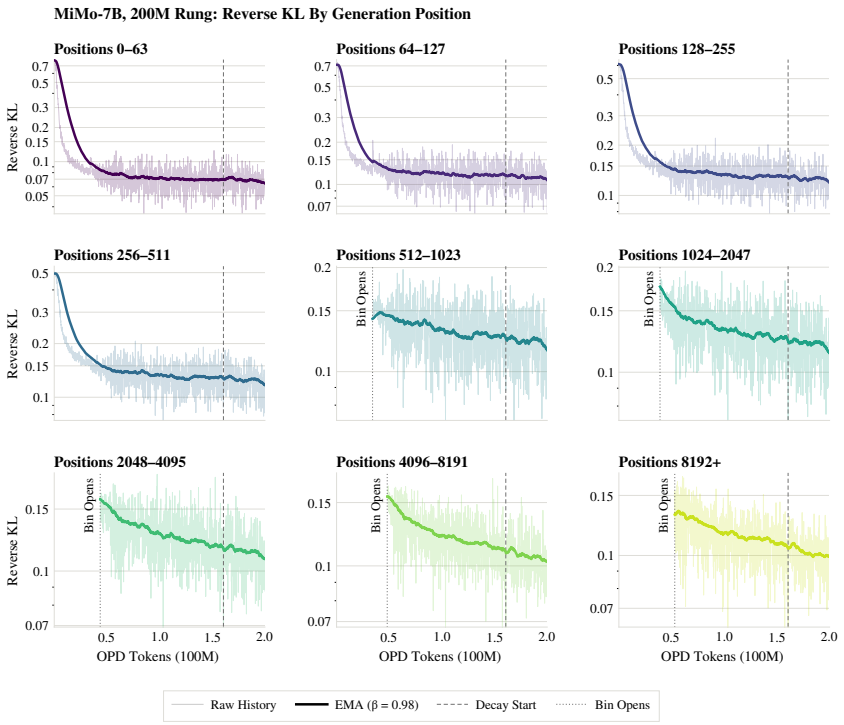


Figure 11: Training curves for position-wise reverse-KL at the first rung of MiMo-7B-RL-0530 training.

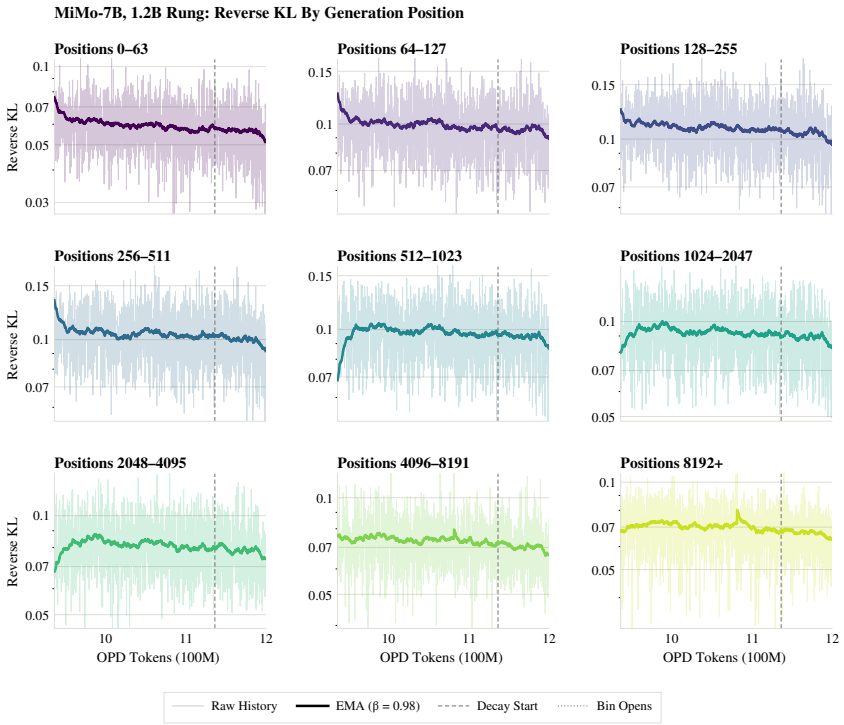


Figure 12: Training curves for position-wise reverse-KL at the run 1B–1.2B OPD tokens of MiMo-7B-RL-0530 training.

C.3 SAMPLE RESPONSES

AIME 2024 I, Problem 13 — Qwen3-4B GDN Hybrid

Problem

Let p be the least prime number for which there exists a positive integer n such that $n^4 + 1$ is divisible by p^2. Find the least positive integer m such that $m^4 + 1$ is divisible by p^2. *Answer: 110*

Model reasoning (thinking mode)

```
Okay, so I need to find the least prime number p such that there's a
positive integer n where n^4 + 1 is divisible by p^2. Then, once I
find that p, I have to find the smallest positive integer m where m^4
+ 1 is divisible by p^2. Alright, let's start by understanding the
problem.

First, the question is about divisibility by p^2. So, for a prime p,
we need to find n such that n^4 == -1 (mod p^2). Then, find the
smallest such prime p, and then find the minimal m for that p.

So, step one: find the smallest prime p where there exists an integer
n with n^4 == -1 (mod p^2).

To approach this, maybe I can start by checking small primes and see
if they satisfy the condition.

First, check p=2. Let's see if there's an n such that n^4 + 1 is
divisible by 4.

n^4 mod 4: Let's compute n^4 mod 4 for n=1,2,3,...

n=1: 1^4 = 1 mod 4, so 1 + 1 = 2 mod 4 != 0.

n=2: 16 mod 4 = 0, so 0 + 1 = 1 mod 4 != 0.

n=3: 81 mod 4 = 1, so 1 + 1 = 2 mod 4 != 0.

n=4: 256 mod 4 = 0, same as n=2. So n^4 + 1 is 1 mod 4. So no solution
for p=2.

Next prime is p=3.

Check if there exists n such that n^4 + 1 == 0 (mod 9).

So n^4 == -1 (mod 9).

Let me compute n^4 mod 9 for n=1 to 8.

n=1: 1

n=2: 16 mod9=7

n=3: 81 mod9=0

n=4: 256 mod9. Let's compute 256 /9: 9*28=252, 256-252=4, so 4.

n=5: 625 mod9. 625 /9=69*9=621, 625-621=4. So 4.

n=6: 1296 mod9=0.

[... 9,743 tokens of reasoning omitted ...]

But since we already confirmed that 110^4 == -1 (mod 289), then 110^4
+ 1 is divisible by 289. Therefore, yes. Therefore, 110 is the answer.
```

```
Therefore, I think the answer is 110. However, I need to check if
there is a mistake in my reasoning. For example, maybe there is a
smaller prime p=17, but for which m is smaller than 110. But according
to our previous check, no. Therefore, I think the answer is 110.

Wait, but let me check if m= 110 is indeed the minimal. Let me check
m= 110 - 289= -179, which is negative, so not allowed. Therefore, yes,
110 is the minimal positive integer.

Therefore, the answer is \boxed{110}.

**Final Answer**
\boxed{110}
```

Model answer

```
To solve the problem, we are given the condition that for a prime
number $ p $, there exists a positive integer $ n $ such that $ n^4 +
1 $ is divisible by $ p^2 $. Our task is to:

1. **Find the smallest such prime $ p $** for which this is true.
2. **Find the smallest positive integer $ m $** such that $ m^4 + 1 $
is divisible by $ p^2 $, where $ p $ is the smallest prime from step
1.

---

### Step 1: Understanding the Condition

We are looking for the **smallest prime** $ p $ such that there exists
an integer $ n $ with:
$$
n^4 + 1 \equiv 0 \pmod{p^2}
$$
This is equivalent to:
$$
n^4 \equiv -1 \pmod{p^2}
$$
This implies that $ -1 $ must be a **quartic residue modulo $ p^2 $**,
or more specifically, that there exists an integer $ n $ such that $
n^4 \equiv -1 \mod p^2 $.

---

### Step 2: Modular Conditions on $ p $

Let's analyze the congruence $ n^4 \equiv -1 \mod p $, which is
necessary for $ n^4 + 1 \equiv 0 \mod p $.

- $ n^4 \equiv -1 \mod p $ implies that $ -1 $ is a **4th power
residue modulo $ p $**.
- This is only possible if $ p \equiv 1 \mod 8 $, because $ -1 $ is a
4th power modulo a prime $ p $ **if and only if** $ p \equiv 1 \mod 8
$.

Thus, the **smallest prime** $ p $ for which this is possible is the
**first prime congruent to 1 mod 8**, which is:

$$
p = 17
$$

---
```

```
### Step 3: Verifying $ p = 17 $

We now check whether there exists an integer $ n $ such that $ n^4 + 1
$ is divisible by $ 17^2 = 289 $.

We first confirm that $ n^4 \equiv -1 \mod 17 $ has solutions. Since
17 is $ 1 \mod 8 $, it is known that $ -1 $ is a 4th power residue
modulo 17. Indeed, we can verify that:

$$
8^4 = 4096 \equiv -1 \mod 17
$$

So $ n = 8 $ is a solution to $ n^4 \equiv -1 \mod 17 $. Now, we use
**Hensel's Lemma** to lift this solution to modulo $ 289 $.

---

### Step 4: Lifting the Solution to Modulo 289

We seek an integer $ n $ such that:
$$
n \equiv 8 \mod 17 \quad \text{and} \quad n^4 \equiv -1 \mod 289
$$

Let $ n = 8 + 17k $. Then:
$$
n^4 = (8 + 17k)^4 = 8^4 + 4 \cdot 8^3 \cdot 17k + \text{(higher powers
of 17)}
$$
Since higher powers of 17 are divisible by $ 289 $, we only need to
consider the first two terms modulo 289.

So:
$$
n^4 \equiv 8^4 + 4 \cdot 8^3 \cdot 17k \mod 289
$$
Compute:
- $ 8^4 = 4096 \equiv 50 \mod 289 $
- $ 8^3 = 512 \equiv 223 \mod 289 $
- So:
$$
n^4 \equiv 50 + 4 \cdot 223 \cdot 17k \mod 289
$$
$$
\Rightarrow n^4 \equiv 50 + 136 \cdot 17k \mod 289
$$
$$
\Rightarrow n^4 \equiv 50 + 136 \cdot 17k \mod 289
$$

We want this to be $ \equiv -1 \mod 289 $, so:
$$
50 + 136 \cdot 17k \equiv -1 \mod 289
\Rightarrow 136 \cdot 17k \equiv -51 \mod 289
$$

Simplify:
- $ 136 \cdot 17 = 2312 \equiv 136 \mod 289 $
- So:
$$
136k \equiv -51 \mod 289
\Rightarrow 136k \equiv 238 \mod 289
$$
```

```
Now solve:
$$
136k \equiv 238 \mod 289
$$

We compute the inverse of 136 modulo 289. Since $ \gcd(136, 289) = 17
$, and 17 divides 238, we can divide the equation by 17:
$$
8k \equiv 14 \mod 17
$$

Find the inverse of 8 modulo 17. Try $ x = 15 $: $ 8 \cdot 15 = 120
\equiv 1 \mod 17 $, so inverse is 15.

So:
$$
k \equiv 14 \cdot 15 \mod 17 \equiv 210 \mod 17 \equiv 6 \mod 17
$$

Therefore:
$$
k = 6 + 17m
\Rightarrow n = 8 + 17k = 8 + 17 \cdot (6 + 17m) = 110 + 289m
$$

So the smallest positive solution is:
$$
n = 110
$$

---

### Step 5: Final Answer

We have shown that:

- The **smallest prime $ p $** for which there exists $ n $ such that
$ n^4 + 1 $ is divisible by $ p^2 $ is $ p = 17 $.
- The **smallest positive integer $ m $** such that $ m^4 + 1 $ is
divisible by $ 17^2 = 289 $ is $ m = 110 $.

---

### Final Answer

$$
\boxed{110}
$$
```

NIAH Multi-Key, 32K Context — Qwen3-4B GDN Hybrid

Prompt

```
Some special magic numbers are hidden within the following text. Make
sure to memorize them. I will quiz you about the numbers afterwards.

[... 14,350 filler sentences: "The grass is green. The sky is blu..."
...]
One of the special magic numbers for spirited-fiddle is: 8350860.
[... 2,075 filler sentences: "The grass is green. The sky is blu..."
...]
One of the special magic numbers for gentle-pillar is: 6351498.
[... 9,050 filler sentences: "The grass is green. The sky is blu..."
...]
```

```
One of the special magic numbers for noble-meadow is: 8411372.
[... 100 filler sentences: "The grass is green. The sky is blu..."
...]
One of the special magic numbers for luminous-ember is: 1144632.
[... 6,800 filler sentences: "The grass is green. The sky is blu..."
...]

What is the special magic number for luminous-ember mentioned in the
provided text?
```

Model output

```
The special magic number for **luminous-ember** mentioned in the text
is **1144632**.
```

AIME 2024 I, Problem 14 — MiMo-7B-RL-0530 GDN Hybrid

Problem

Let $ABCD$ be a tetrahedron such that $AB = CD = \sqrt{41}$, $AC = BD = \sqrt{80}$, and $BC = AD = \sqrt{89}$. There exists a point I inside the tetrahedron such that the distances from I to each of the faces of the tetrahedron are all equal. This distance can be written in the form $\frac{m\sqrt{n}}{p}$, where m, n, and p are positive integers, m and p are relatively prime, and n is not divisible by the square of any prime. Find $m+n+p$. *Answer: 104*

Model reasoning (thinking mode)

```
Okay, so I have this problem about a tetrahedron ABCD with specific
edge lengths, and I need to find the distance from an interior point I
to each face, which is equal. The answer should be in the form
m*sqrt(n)/p, and then find m + n + p. Hmm, let's start by
understanding the problem step by step.

First, the tetrahedron has edges AB = CD = sqrt(41), AC = BD =
sqrt(80), and BC = AD = sqrt(89). The point I is inside the
tetrahedron and is equidistant from all four faces. That distance is
called the inradius, right? Wait, but in a tetrahedron, the inradius
is the radius of the sphere tangent to all four faces. But the problem
says there exists such a point I, so maybe this tetrahedron is
isohedral? Or maybe it's a special tetrahedron where such a point
exists. Maybe it's a tetrahedron that's symmetric enough to have an
inradius.

But how do I calculate the inradius? For a tetrahedron, the formula
for the inradius r is given by r = 3V / S, where V is the volume of
the tetrahedron and S is the total surface area. Wait, is that right?
Let me recall. For a tetrahedron, the volume can be calculated using
the scalar triple product, but maybe there's a more straightforward
way here given the edge lengths.

[... 8,692 tokens of reasoning omitted ...]

Wait, vectors BC = C - B = (4-4, 0-5, 8-0) = (0, -5, 8)

Vectors BD = D - B = (0-4, 5-5, 8-0) = (-4, 0, 8)

Cross product BC x BD is determinant:

|i  j  k|
|0 -5  8|
|-4 0  8|

= i*(-5*8 - 8*0) - j*(0*8 - 8*(-4)) + k*(0*0 - (-5)*(-4))
```

```
= i*(-40 - 0) - j*(0 + 32) + k*(0 - 20)

= (-40, -32, -20)

Same as before. So magnitude is sqrt(1600 + 1024 + 400) = sqrt(3024) =
12*sqrt(21). Area is half that, so 6*sqrt(21). Correct.

So, all faces are indeed 6*sqrt(21), total surface area 24*sqrt(21).
Volume 160/3. So, r = 3*(160/3)/(24*sqrt(21)) = 160/(24*sqrt(21)) =
20/(3*sqrt(21)) = 20*sqrt(21)/63. Correct.

Therefore, the answer is 20 + 21 + 63 = 104.

**Final Answer**
\boxed{104}
```

Model answer

```
To solve the problem, we start by noting that the tetrahedron \(ABCD\)
has edge lengths \(AB = CD = \sqrt{41}\), \(AC = BD = \sqrt{80}\), and
\(BC = AD = \sqrt{89}\). We need to find the inradius of this
tetrahedron, which is the distance from the interior point \(I\) to
each face, given in the form \(\frac{m \sqrt{n}}{p}\).

First, we recognize that the tetrahedron can be embedded in a
rectangular box with dimensions \(4 \times 5 \times 8\). This is
confirmed by solving the system of equations derived from the edge
lengths:
\[
x^2 + y^2 = 41, \quad x^2 + z^2 = 80, \quad y^2 + z^2 = 89.
\]
Solving these, we find \(x = 4\), \(y = 5\), and \(z = 8\).

Next, we calculate the volume \(V\) of the tetrahedron using the
scalar triple product. The vectors from point \(A(0,0,0)\) to
\(B(4,5,0)\), \(C(4,0,8)\), and \(D(0,5,8)\) are \(\overrightarrow{AB}
= (4,5,0)\), \(\overrightarrow{AC} = (4,0,8)\), and
\(\overrightarrow{AD} = (0,5,8)\). The cross product
\(\overrightarrow{AB} \times \overrightarrow{AC}\) is:
\[
\overrightarrow{AB} \times \overrightarrow{AC} = \begin{vmatrix}
\mathbf{i} & \mathbf{j} & \mathbf{k} \\
4 & 5 & 0 \\
4 & 0 & 8
\end{vmatrix} = (40, -32, -20).
\]
The dot product with \(\overrightarrow{AD}\) is:
\[
(40, -32, -20) \cdot (0, 5, 8) = 0 - 160 - 160 = -320.
\]
The magnitude of this vector is \(\sqrt{40^2 + (-32)^2 + (-20)^2} =
\sqrt{1600 + 1024 + 400} = \sqrt{3024} = 12\sqrt{21}\). The volume is:
\[
V = \frac{1}{6} \left| -320 \right| = \frac{320}{6} = \frac{160}{3}.
\]

The total surface area \(S\) is calculated by finding the area of each
face. Each face is a triangle with area \(\frac{1}{2}\) the magnitude
of the cross product of two edges. For example, for face \(ABC\), the
vectors are \(\overrightarrow{AB} = (4,5,0)\) and
\(\overrightarrow{AC} = (4,0,8)\), and the cross product is \((40,
-32, -20)\). The magnitude is \(\sqrt{40^2 + (-32)^2 + (-20)^2} =
12\sqrt{21}\), so the area is \(\frac{1}{2} \times 12\sqrt{21} =
6\sqrt{21}\). Since all four faces are congruent, the total surface
area is:
```

```
\[
S = 4 \times 6\sqrt{21} = 24\sqrt{21}.
\]

The inradius \(r\) is given by the formula \(r = \frac{3V}{S}\).
Substituting the values, we get:
\[
r = \frac{3 \times \frac{160}{3}}{24\sqrt{21}} =
\frac{160}{24\sqrt{21}} = \frac{20}{3\sqrt{21}} =
\frac{20\sqrt{21}}{63}.
\]

Thus, the inradius is \(\frac{20\sqrt{21}}{63}\). Here, \(m = 20\),
\(n = 21\), and \(p = 63\). These values are coprime and \(n\) is
square-free. Therefore, \(m + n + p = 20 + 21 + 63 = 104\).

The final answer is:
\[
\boxed{104}
\]
```

NIAH Multi-Query, 32K Context — MiMo-7B-RL-0530 GDN Hybrid

Prompt

```
Some special magic numbers are hidden within the following text. Make
sure to memorize them. I will quiz you about the numbers afterwards.

[... 12,525 filler sentences: "The grass is green. The sky is blu..."
...]
One of the special magic numbers for vivid-walnut is: 8616278.
[... 4,900 filler sentences: "The grass is green. The sky is blu..."
...]
One of the special magic numbers for placid-lantern is: 7006521.
[... 1,375 filler sentences: "The grass is green. The sky is blu..."
...]
One of the special magic numbers for mellow-bramble is: 8084209.
[... 4,600 filler sentences: "The grass is green. The sky is blu..."
...]
One of the special magic numbers for rugged-ember is: 5916232.
[... 8,950 filler sentences: "The grass is green. The sky is blu..."
...]

What are the special magic numbers for vivid-walnut, placid-lantern,
mellow-bramble, and rugged-ember mentioned in the provided text? State
each key with its number.
```

Model output

```
The special magic numbers mentioned in the text are as follows:

- **vivid-walnut**: 8616278
- **placid-lantern**: 7006521
- **mellow-bramble**: 8084209
- **rugged-ember**: 5916232

Each key is paired with its corresponding number as stated in the
text.
```