WHEN TO THINK FAST AND SLOW? AMOR: ADAPTIVE ENTROPY GATE FOR HYBRID MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recurrent-attention hybrids aim to combine the efficiency of recurrence with the contextual recall of attention, but existing approaches typically apply attention uniformly across all positions, even when the recurrent state alone is sufficient for accurate prediction. We introduce AMOR (*Adaptive Metacognitive Output Router*), a post-hoc hybrid architecture that selectively invokes attention based on predictive uncertainty. A recurrent backbone is augmented with entropy-gated attention blocks that activate only when the model's output entropy exceeds a dynamic threshold derived from a running batch median and scaled standard deviation. The resulting binary gate requires no learned routing parameters. Pretrained from scratch on FineWeb-Edu and with attention invoked on only $\sim 40\%$ of positions, one of the AMOR variants (Mamba2 or Gated DeltaNet backbones) achieves the highest eight-task common-sense reasoning average at each scale among pure recurrent, pure attention, and fixed-schedule hybrid models. AMOR also improves retrieval performance over pure recurrent models while remaining competitive against fixed-schedule hybrids. Additionally, AMOR retains the long-context robustness of its recurrent backbones, where the Transformer and other hybrid architectures degrade under distribution shift. These results suggest that *when* attention is applied matters as much as *how much*: selectively allocating attention based on predictive uncertainty improves accuracy, robustness, and efficiency, offering a simple alternative to uniform or fixed routing strategies.

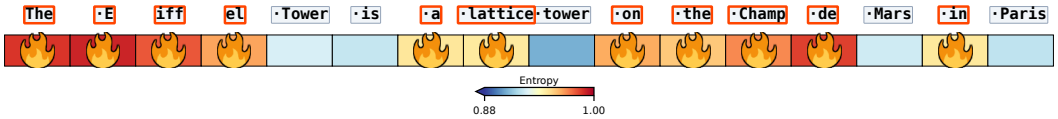

Figure 1: **Entropy gate firing pattern of AMOR** on the input sentence "The Eiffel Tower is a lattice tower on the Champ de Mars in Paris." The example illustrates how AMOR combines the strengths of recurrent models and attention: the gate fires early in the sequence, when uncertainty over possible continuations is high, and again on rarer or information-dense tokens such as *lattice* and *Champ*. In contrast, the gate remains inactive on highly predictable tokens that can be inferred from local context or world knowledge (e.g., *Tower*, *Paris*, conditioned on *Eiffel*). This suggests that **AMOR selectively deploys attention when next-token prediction becomes difficult**, while relying on the lightweight recurrent backbone for easier continuations.

1 INTRODUCTION

Since GPT-3 (Brown et al., 2020), attention (Vaswani et al., 2017) has been the de facto sequence mixer for language models: every token attends to all preceding tokens, enabling fully parallel training. This uniform pairwise coupling, however, incurs $O(N^2)$ training cost and requires a key-value cache at inference that grows linearly with sequence length, making long-context generation increasingly expensive. To address this, linear recurrent models replace quadratic attention with a fixed-size state: each layer selectively projects the input into a bounded representation, enabling constant memory and $O(1)$ time per token at decode (Lahoti et al., 2026; Yang et al., 2025). However, this efficiency comes with a structural limitation. **A finite state cannot faithfully store an unbounded set of distinct**

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.

key-value associations, making in-context retrieval inherently lossy and leading to degradation on exact-copy stress tests[1] (Arora et al., 2024a; Jelassi et al., 2024; Arora et al., 2024b).

Hybrid architectures combine the two: attention handles recall while the recurrent backbone keeps long-context decode cheap. Open hybrid models such as Kimi K3 (Kimi Team, 2026) and Nemotron 3 Ultra (NVIDIA, 2026b) are among the strongest open-weight systems (Kimi K3 is also competitive with closed frontier models on standard benchmarks). However, these hybrid models still rely on predetermined layer schedules, applying attention uniformly over all tokens, incurring **full quadratic cost even when the recurrent state alone is sufficient to predict the next token**.

In this work, we aim to **resolve the tension between computationally heavy attention and compact recurrent models by selectively allocating expensive computation**. We propose AMOR (Adaptive Metacognitive Output Router; see Fig. 1), a *post-hoc* hybrid that introduces a lightweight, human-inspired gating mechanism. Motivated by dual-process accounts of cognition (Kahneman, 2011)[2], where fast automatic responses are complemented by slower deliberation when uncertainty is high, AMOR preserves a complete recurrent backbone and appends K entropy-gated attention blocks. At each block, AMOR computes the normalized prediction entropy of each token in the input and activates the block only when entropy exceeds an adaptive threshold. The threshold is defined as an exponential moving average of the batch median with a small standard-deviation-scaled offset, allowing it to track shifts in the backbone's uncertainty distribution during training. This produces an intuitive and stable routing rule that requires only the model's own predictive uncertainty and no additional learned routing parameters.

We pretrain eight architectures at three scales (180M, 440M, and 1.5B) under a uniform Chinchilla-optimal token budget. Across scales, AMOR attends to $\sim 40\%$ of positions during training. At inference, firing rates vary by domain but remain stable across context lengths, even beyond those seen in training (Table 8).

Despite sparse attention, an AMOR variant achieves the best average common-sense reasoning score at all scales (Fig. 4), and outperforms its recurrent backbone on retrieval while remaining competitive with fixed-schedule hybrid baselines (Fig. 5). Furthermore, AMOR preserves its backbone's robustness on long-context tasks, where both Transformer and other hybrids degrade under out-of-distribution vanilla RoPE (Su et al., 2021) (Fig. 6). Overall, by routing attention only to uncertain positions, AMOR augments its recurrent backbone. Our results suggest that efficiency and expressivity are not opposing forces, but a routing problem of **where and when to allocate compute**.[3]

2 RELATED WORK

Recurrent models. Linear recurrent models offer efficient alternatives to attention, including S4 (Gu et al., 2022), Mamba (Gu & Dao, 2024), Mamba2 (Dao & Gu, 2024), Mamba3 (Lahoti et al., 2026), DeltaNet (Yang et al., 2024), Gated DeltaNet (Yang et al., 2025), and RWKV (Peng et al., 2023). AMOR is backbone-agnostic: any recurrent mixer can serve as the unmodified backbone before the entropy gates. We study Mamba2 and Gated DeltaNet.

By compressing past context into a fixed-size state, recurrent models support linear-time training and $O(1)$ updates per decoded token (Dao & Gu, 2024; Yang et al., 2025), versus $O(N)$ for full attention over N tokens. However, this fixed capacity limits how much information they can retain for later retrieval. Jelassi et al. (2024) show that recurrent models with fixed-size states fail to copy sequences whose information content exceeds their state size, and that pretrained Mamba underperforms Pythia Transformers on synthetic recall tasks such as copying and phone-book lookup. Consistent with this, Arora et al. (2024a) report a substantial recall-accuracy gap between attention-based models and Mamba on real-world, recall-intensive benchmarks, and Arora et al. (2024b) extend these findings to large-scale retrieval settings, where Transformers retain a clear advantage even at a billion-parameter scale. Together, these results suggest that **while recurrent models offer compelling efficiency gains, they struggle to match the flexible, content-addressable memory afforded by attention.**

[1] Exact-copy stress tests evaluate a model's ability to reproduce a target span verbatim from context, probing lossless storage and retrieval rather than generalization.

[2] "System 1 operates automatically and quickly, with little or no effort and no sense of voluntary control. System 2 allocates attention to the effortful mental activities that demand it, including complex computations".

[3] Code: `https://anonymous.4open.science/r/amor/`.

Recurrent-attention hybrids. To address the limitations of attention and recurrent models, recent work has explored hybrid architectures, which we divide into *serial* and *fused* hybrids.

Serial hybrids replace the recurrent mixer with attention at a *fixed subset* of layers. Mamba-based hybrids typically use sparse schedules: Jamba (Lieber et al., 2024) uses a 1:7 attention-to-Mamba ratio, Bamba (Chu et al., 2024) ∼1:10, and Nemotron 3 Super (NVIDIA, 2026a) 1:5. Delta-rule-based hybrids tend toward denser schedules, with Qwen3.5 (Qwen Team, 2026), Kimi Linear (Kimi Team, 2025), and Olmo Hybrid (Merrill et al., 2026) near 1:3. Other examples include Samba (Ren et al., 2025), Griffin (De et al., 2024), Zamba2 (Glorioso et al., 2024), Granite 4.0 (Soule & Bergmann, 2025), and Jet-Nemotron (Gu et al., 2025). Fused hybrids instead run attention *in parallel* with the recurrent mixer: Hymba (Dong et al., 2025) averages attention and Mamba heads with learned per-channel scales, while Falcon-H1 (Zuo et al., 2025) concatenates them at a fixed channel ratio.

Prior hybrids thus focus primarily on *depth*: determining **where** and in what proportion to place attention. Bae et al. (2025) find that attention distributed throughout the network improves fused hybrids, while serial hybrids favor an attention-to-Mamba ratio of $\sim$ 1:5. Similarly, Wang et al. (2025) find that full-attention-to-linear ratios of 1:3–1:6 recover Transformer-level recall.

In both serial and fused hybrids, attention placement is fixed at design time. AMOR instead acts on **when** attention fires, not just **where** or **how much**. It appends K post-hoc attention blocks after the recurrent backbone and selectively activates them at each token via normalized-entropy gating.

Conditional Compute. Soft methods such as ACT (Graves, 2016) and PonderNet (Banino et al., 2021) use differentiable halting, while hard-routing methods make discrete compute decisions. Mixture-of-Depths (MoD) (Raposo et al., 2024) routes a fixed token budget through attention and MLP blocks, while DeepSeek Sparse Attention (DeepSeek-AI, 2025), Native Sparse Attention (Yuan et al., 2025), and MoBA (Lu et al., 2025) prune attention via top-k routing.

Most conditional-compute methods route based on *input* representations, such as MoD's router signals or MoBA's query-key similarity. A smaller line of work conditions on the model's *output*; CALM (Schuster et al., 2022), for example, uses the top-1 vs. top-2 probability margin for layer-wise early exit. AMOR instead uses the full next-token distribution's entropy for per-position gating *within* attention sublayers, selectively invoking attention rather than skipping layers.

3 AMOR

We now describe AMOR in detail, starting with the general architecture (Section 3.1), and focusing on our entropy gate (Section 3.2) and each block's flow (Section 3.3) during training (Section 3.4) and inference (Section 3.5). An optional distilled approximation to the entropy gate is evaluated separately as a deployment ablation in Section 5; its definition is in Appendix G.

3.1 ARCHITECTURE

We instantiate AMOR within a standard recurrent-attention hybrid skeleton:

$$\text{Embed} \to \left[\text{Recurrent Mixer}_i + \text{SwiGLU-MLP}_i\right]_{i=1}^{N} \to \left[\text{AMOR Block}_\ell\right]_{\ell=0}^{K-1} \to \text{final norm} \to \text{LM Head}. \tag{1}$$

Here N is the backbone depth, counting recurrent-mixer–MLP layers indexed by i. We index AMOR blocks by ℓ, use K for their total count, and reserve κ for the entropy-threshold offset multiplier.

We study two variants that differ only in the recurrent mixer: AMOR-Mamba2 (based on Mamba2) and AMOR-Gated DeltaNet (based on Gated DeltaNet). Both insert $K{=}3$ entropy-gated AMOR blocks between the backbone and the final LM head (Fig. 2).[4]

With entropy gating, each model reuses a single LM head, weight-tied to the embedding, invoked $K{+}1$ times per forward pass: once on each AMOR block's normalized input to compute its gate (Section 3.2), and once on the final normalized residual to produce output logits.

[4] $K{=}3$ matches the attention budget of our serial and fused hybrid baselines: three attention layers in a 24-layer model, following Jamba's (Lieber et al., 2024) recommended 1:7 ratio.

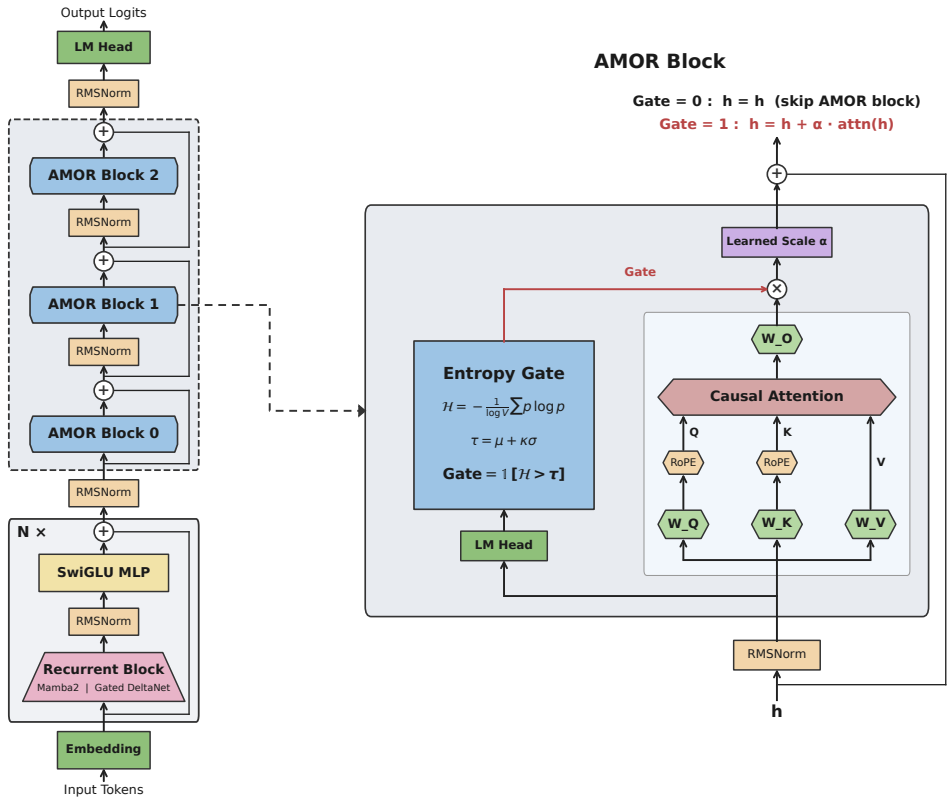

Figure 2: **AMOR architecture.** A recurrent backbone (Mamba2 or Gated DeltaNet, with SwiGLU MLPs) feeds three AMOR blocks. Each block queries the tied LM head, gates causal RoPE attention via normalized entropy, and adds a per-channel $\boldsymbol{\alpha}$-scaled residual update. Block 0 consumes the backbone's terminal norm_f output (no pre-norm); Blocks 1–2 use RMSNorm. $\mathbf{W}_O$ is zero-initialized, so AMOR matches the backbone at initialization.

Within each AMOR block, attention is implemented as causal SDPA over $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ projections of the normalized input, with RoPE (Su et al., 2021) applied to $\mathbf{Q}$ and $\mathbf{K}$. For model dimension $D = d_{\text{model}}$, each block introduces $\sim 4D^2$ parameters ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$ plus a per-channel scaling vector $\boldsymbol{\alpha}$).

A key design choice is that the three AMOR blocks are intentionally *asymmetric*. Block 0 operates directly on the backbone's final norm_f-normalized residual (the LM-ready representation) and applies attention refinement on this post-norm_f stream. In contrast, Blocks 1 and 2 follow a standard pre-norm formulation: each first applies its own RMSNorm (Zhang & Sennrich, 2019) before computing the entropy gate and attention, then updates the (unnormalized) residual. This distinction is important because the residual is no longer unit-RMS scaled after Block 0's intervention. This asymmetry enforces a key invariant: the recurrent backbone remains a complete, standalone LM, rather than the first stage of a deeper stack. AMOR thus acts strictly as a conditional refinement layer on top of an already valid LM representation, aligning with the dual-process framing in Section 1. We also considered a symmetric alternative in which norm_f is repurposed as Block 0's pre-norm and the residual remains unnormalized throughout the AMOR stack (Appendix H).

Lastly, each AMOR block adds a per-channel $\boldsymbol{\alpha}$-scaled attention update to the residual. The per-channel scaling vector $\boldsymbol{\alpha} \in (0, 1)^D$ allows each dimension to control its own contribution, rather than relying on a single global factor. After accumulating K such updates, a single $\mathrm{final\,norm}$ rescales the stream before the LM head. To preserve the backbone's initial behavior, $\mathbf{W}_O$ is zero-initialized, making AMOR functionally identical to the recurrent backbone at initialization; deviations are learned progressively as $\mathbf{W}_O$ updates. Finally, AMOR blocks contain no MLP (Appendix H), isolating the effect of attention-based refinement.

3.2 ENTROPY GATE

Each AMOR block decides where to engage attention based on the backbone's predictive uncertainty. At token position t, given the block's normalized input $\tilde{h}_t$, the shared LM head produces logits $\mathbf{z}_t = \mathrm{lm_head}(\tilde{h}_t) \in \mathbb{R}^V$ over the V=128,256-token Llama-3.1 vocabulary. From these logits, we compute the normalized output entropy:

$$\mathcal{H}_t \;=\; -\frac{1}{\log V}\sum_{v=1}^{V} p_{t,v}\,\log p_{t,v} \;\in [0,1], \qquad \mathbf{p}_t = \mathrm{softmax}(\mathbf{z}_t). \tag{2}$$

Here $p_{t,v}$ is the predicted probability of vocabulary token v at position t.

Logits are detached before entropy computation: $\mathcal{H}_t$ drives the gating decision and does not backpropagate into the backbone, keeping the gate purely diagnostic. At each training forward pass, the block updates two non-learnable EMA buffers from its entropies $\mathcal{H}^{\text{batch}}$ over all positions in the batch,

$$\mu \;\leftarrow\; (1-\eta)\,\mu \;+\; \eta\,\mathrm{median}(\mathcal{H}^{\text{batch}}), \qquad \sigma \;\leftarrow\; (1-\eta)\,\sigma \;+\; \eta\,\mathrm{std}(\mathcal{H}^{\text{batch}}), \tag{3}$$

with η=0.01 (momentum 0.99), and forms the threshold

$$\tau \;=\; \mu \;+\; \kappa\,\sigma, \qquad \kappa = 0.2. \tag{4}$$

The gate is binary: $g_t = \mathbf{1}[\mathcal{H}_t > \tau] \in \{0,1\}$. The statistics μ and σ are maintained as non-learnable EMA buffers and frozen at inference; accordingly, the threshold τ adapts to the entropy distribution during training and remains fixed at evaluation. We set $\tau = \mu + \kappa\sigma$, where using the median-based μ provides robustness to per-batch outliers, and scaling by σ stabilizes the firing rate across model sizes. In contrast, a fixed offset leads to drift, as σ typically shrinks with improved calibration. In practice, we target a per-block firing rate of $\sim 40\%$. A single choice of κ=0.2 achieves this consistently across three model scales and two backbones (Mamba2 and Gated DeltaNet), with firing rates remaining stable throughout training (Appendix D).

3.3 BLOCK FORWARD AND GRADIENT FLOW

Within block ℓ, let $h_t^{(\ell)}$ be the input residual and $\tilde{h}_t^{(\ell)}$ its normalized copy used for gating and attention. We set $\tilde{h}_t^{(0)} = h_t^{(0)}$ (the backbone's norm_f output), while for $\ell \geq 1$, $\tilde{h}_t^{(\ell)} = \mathrm{RMSNorm}_\ell\Big(h_t^{(\ell)}\Big)$, with $h_t^{(\ell)}$ the unnormalized residual from block $\ell-1$. Here $\tilde{h}$ denotes the whole normalized input sequence. Training uses dense attention with a post-mask ($\odot$ denotes elementwise multiplication):

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} \;=\; \mathbf{W}_Q\,\tilde{h},\;\; \mathbf{W}_K\,\tilde{h},\;\; \mathbf{W}_V\,\tilde{h}, \tag{5}$$

$$(\mathbf{Q}, \mathbf{K}) \;\leftarrow\; \mathrm{RoPE}(\mathbf{Q}, \mathbf{K}), \tag{6}$$

$$\mathbf{a}_t \;=\; \mathbf{W}_O\,\mathrm{SDPA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_t, \tag{7}$$

$$h_t^{(\ell+1)} \;=\; h_t^{(\ell)} \;+\; g_t \cdot \boldsymbol{\alpha} \odot \mathbf{a}_t. \tag{8}$$

We parameterize the per-channel scaling vector as $\boldsymbol{\alpha} = \mathrm{sigmoid}(\tilde{\boldsymbol{\alpha}})$ with $\tilde{\boldsymbol{\alpha}}_{\text{init}}$=$\mathbf{0}$, yielding $\boldsymbol{\alpha}_{\text{init}} = 0.5$ in every channel.[5] The gate is binary and non-differentiable ($\partial g_t / \partial \mathcal{H}_t = 0$ almost everywhere), ensuring that no gradients flow through the threshold or back into the backbone via $\mathcal{H}$.

Gradient flow is position-dependent (Fig. 3). At firing positions ($g_t = 1$), all parameters $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O, \boldsymbol{\alpha}$ receive gradients as in standard attention. At non-firing positions ($g_t = 0$), gradients to $\mathbf{W}_Q, \mathbf{W}_O$, and $\boldsymbol{\alpha}$ vanish at t, while $\mathbf{W}_K$ and $\mathbf{W}_V$ still receive gradients via causal attention from later firing positions $j > t$. This asymmetry is intentional: $\mathbf{Q}$ is only needed where the gate fires, whereas $\mathbf{K}$ and $\mathbf{V}$ must represent all positions for future retrieval.

3.4 TRAINING MODE

We train in a dense $\mathrm{full_with_mask}$ mode (causal SDPA at all positions, masked by the gate), which is mathematically equivalent to the $\mathrm{true_sparse}$ inference implementation; we use the dense form for efficiency via fused FlashAttention-2 kernels (Dao, 2024) (Appendix C.1).

[5] As a pretraining safeguard, we clamp pre-sigmoid parameters at -1 ($\boldsymbol{\alpha} \geq \mathrm{sigmoid}(-1) \approx 0.269$). Across inspected final checkpoints, all channels in Blocks 0–1 remained above the floor; only 5/2,048 channels (0.2%) in 1.5B AMOR-Gated DeltaNet's Block 2 were at it.

3.5 INFERENCE: CONDITIONAL SKIP AND KV CONTINUITY

At decode time, the gate is evaluated per token and induces a conditional skip: when $g_t = 0$, the $\mathbf{W}_Q$ projection, RoPE on $\mathbf{Q}$, SDPA, and $\mathbf{W}_O$ are bypassed. In all cases, $\mathbf{W}_K$, $\mathbf{W}_V$, RoPE on $\mathbf{K}$, and KV cache updates are performed, ensuring that future firing queries can attend to all past positions, including non-firing ones.

With the threshold frozen, AMOR yields *difficulty-adaptive compute*: **attention is invoked more frequently when the backbone is uncertain and less when it is confident**.

For the native entropy gate, counting attention matmul savings only, gating reduces matrix-multiply FLOPs relative to otherwise identical always-on attention when

$$(1 - f) \cdot C_{\text{attn}}(L) \;>\; C_{\text{lm_head}}, \qquad (9)$$

where f= firing rate and L=attended cache length including the current token. For one decoded token in one AMOR block, $C_{\text{attn}}(L) = 4DL$ and $C_{\text{lm_head}} = 2VD$. This is an arithmetic condition, not a latency guarantee, as memory traffic, kernel execution, and synchronization also affect runtime. Appendix F.5 derives the entropy gate's arithmetic threshold and compares routing costs; Section 5 reports measured latency and memory at 1.5B.

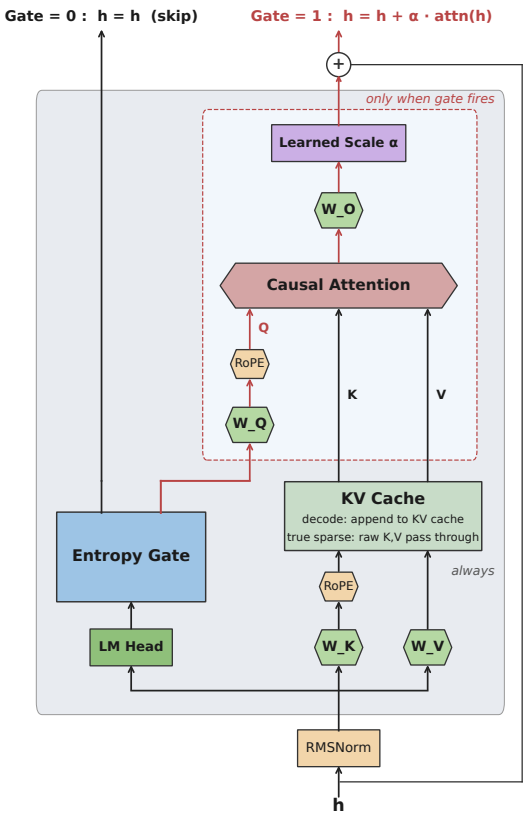

Figure 3: **AMOR decode path.** When g_t=0, the block skips $\mathbf{W}_Q$, RoPE on $\mathbf{Q}$, SDPA, and $\mathbf{W}_O$. $\mathbf{W}_K$, $\mathbf{W}_V$, RoPE on $\mathbf{K}$, and KV-cache updates always run.

4 EXPERIMENTS

We compare AMOR with pure recurrent, pure attention, and fixed-schedule hybrid baselines to assess whether selective attention allocation improves downstream performance. The eight architectures all use sequence mixers and SwiGLU MLPs, with differences in mixer type and attention placement.

Baselines. **Backbone models** include pure recurrent mixers (Mamba2 (Dao & Gu, 2024), Gated DeltaNet (Yang et al., 2025)) and a pure attention baseline (a Llama-style Transformer (Grattafiori et al., 2024) with full RoPE (Su et al., 2021)). **Fixed-schedule hybrids** combine recurrence with attention at predetermined layers. We evaluate two serial hybrids (Mamba2, Gated DeltaNet), which replace recurrent blocks with attention at layers $\{4, 8\}$ for N=12 and $\{6, 12, 18\}$ for N=24, following Jamba (Lieber et al., 2024). We also include a fused hybrid (Mamba2), which runs recurrent and attention modules in parallel at layers $\{0, 6, 11\}$ for N=12 and $\{0, 12, 23\}$ for N=24, following Hymba (Dong et al., 2025). We omit a fused Gated DeltaNet variant due to the lack of an open-weight implementation. **Post-hoc hybrids** correspond to AMOR (AMOR-Mamba2, AMOR-Gated DeltaNet), which append K=3 entropy-gated attention blocks to an otherwise unmodified backbone (Fig. 2).

Controlled Design. All models are trained from scratch on FineWeb-Edu (Penedo et al., 2024) with the Llama-3.1 tokenizer (Grattafiori et al., 2024) at sequence length T=3,072, using an identical optimization recipe (Appendix C). At each scale, all models are trained on the same Chinchilla-optimal token budget (Hoffmann et al., 2022): 3.69B / 8.97B / 30.7B tokens at 180M / 440M / 1.5B parameters. We follow the high-level designs of Jamba and Hymba while removing orthogonal components to isolate the effect of *fixed-schedule attention*. All hybrids use full causal attention and

the same corresponding Mamba2 or Gated DeltaNet backbone, ensuring consistent attention and recurrent primitives across models (see full details in Appendix B).

Tasks. Following the evaluation protocol of Yang et al. (2025), we evaluate on: (a) common-sense reasoning at 180M, 440M, and 1.5B (Section 4.1); (b) in-context retrieval at 1.5B (Section 4.2); (c) long-context behavior at 1.5B (Section 4.3). This follows the largest-scale evaluation design of Gated DeltaNet (1.3B) and Mamba3 (1.5B) (Yang et al., 2025; Lahoti et al., 2026); Appendix G additionally evaluates the smaller-scale deployment ablation. We use the LM Evaluation Harness (Biderman et al., 2024); task details are in Appendix E, with per-task results in Tables 1–3.

4.1 Common-Sense Reasoning

We evaluate zero-shot accuracy on LAMBADA (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC-Easy, ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), OpenBookQA (Mihaylov et al., 2018) and TruthfulQA-mc2 (Lin et al., 2022). **An AMOR variant outperforms common-sense reasoning on every scale** (Fig. 4; Table 1). Error bars are constructed from five evaluations of the final model checkpoint and show the sample SD. Results across seeds controlling model initialization and training-data order at 180M are reported in Appendix A.2.

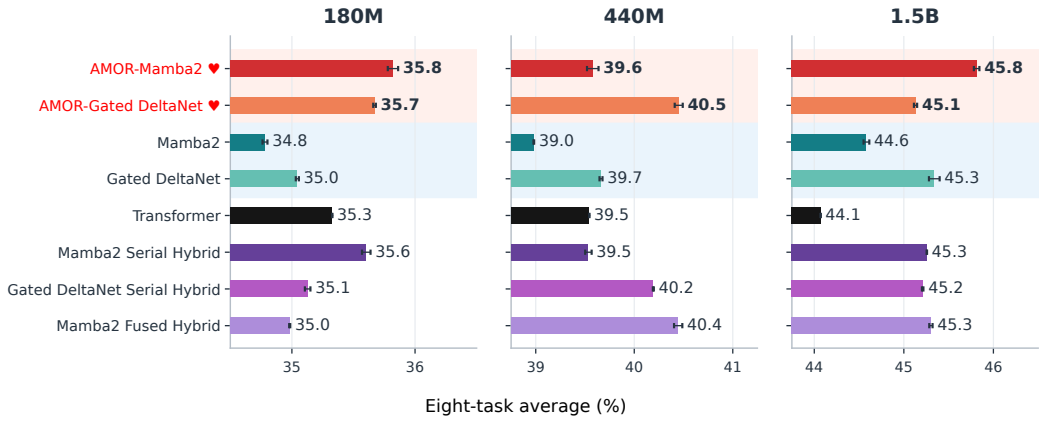

Figure 4: **An AMOR variant outperforms common-sense reasoning on every scale.** Eight-task common-sense reasoning accuracy (detailed scores in Table 1).

4.2 In-Context Retrieval

Retrieval is where attention is least substitutable: cloze tasks require recovering spans from earlier in the input. Pure recurrent backbones underperform sharply, while fixed-schedule hybrids lead. **Both AMOR variants substantially improve over their backbones and remain competitive with fixed-schedule hybrids**, with gains in 11 of 12 cloze retrieval task-backbone comparisons. On Single Needle-in-a-Haystack (S-NIAH), AMOR's advantage is especially evident at 2K context (Fig. 5). Task details and full results are in Appendix E.2 and Table 2.

AMOR achieves these gains with attention restricted to uncertain positions. Zeroing attention outputs reduces the average over SWDE (Hao et al., 2011), SQuAD-completion (Rajpurkar et al., 2016), and FDA (Arora et al., 2023) from 30.4% to 9.0% for AMOR-Mamba2 and 36.4% to 11.4% for AMOR-Gated DeltaNet (Table 25), demonstrating the contribution of its gated attention updates. The Transformer's steep S-NIAH decline at 4K occurs beyond its 3,072-token training context, consistent with vanilla RoPE's poor length extrapolation (Peng et al., 2024); see Section 4.3.

4.3 Long-Context Behavior

LongBench evaluates long-context understanding beyond training. Pure recurrent backbones, which do not explicitly encode position, degrade gracefully, while the pure attention model and fixed-

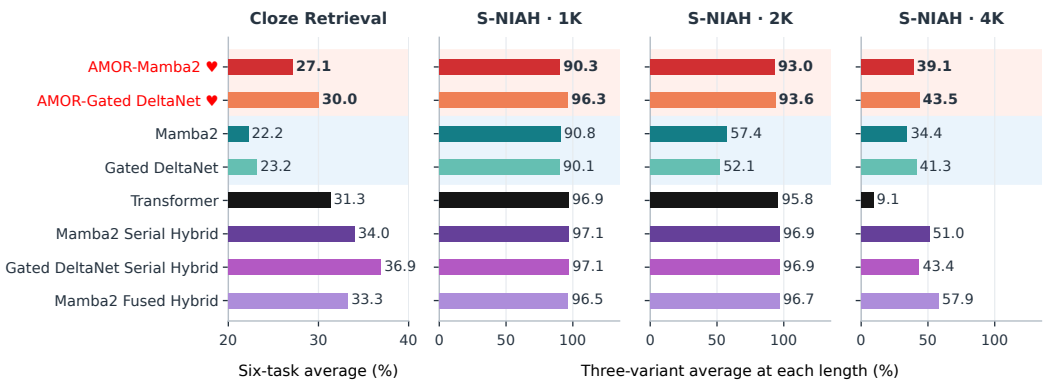

Figure 5: **AMOR improves in-context retrieval over recurrent backbones and remains competitive with fixed-schedule hybrids.** Left: six-task cloze retrieval average. Right: average across three Single Needle-in-a-Haystack (S-NIAH) variants at varying context lengths. 1.5B; full task scores are in Table 2.

schedule hybrids, especially serial, use full RoPE attention without length extension and fall below recurrent baselines. This pattern is reaffirmed on length extrapolation tasks (Fig. 8).

AMOR stays within 0.5 points of its recurrent backbone on LongBench and outperforms the Transformer and fixed-schedule hybrids (Fig. 6, left; Table 3). On NarrativeQA, AMOR has lower perplexity at 20K than at the 3K training context, while the Transformer and serial hybrids deteriorate sharply (Fig. 6, right).

Our intuition is that AMOR mitigates RoPE distribution shift via its entropy gate: when quiet, the gate skips attention, predicting via the RoPE-free recurrent backbone. This is consistent with stable long-context firing rates within domains (Table 8), though causal conclusions require future work.

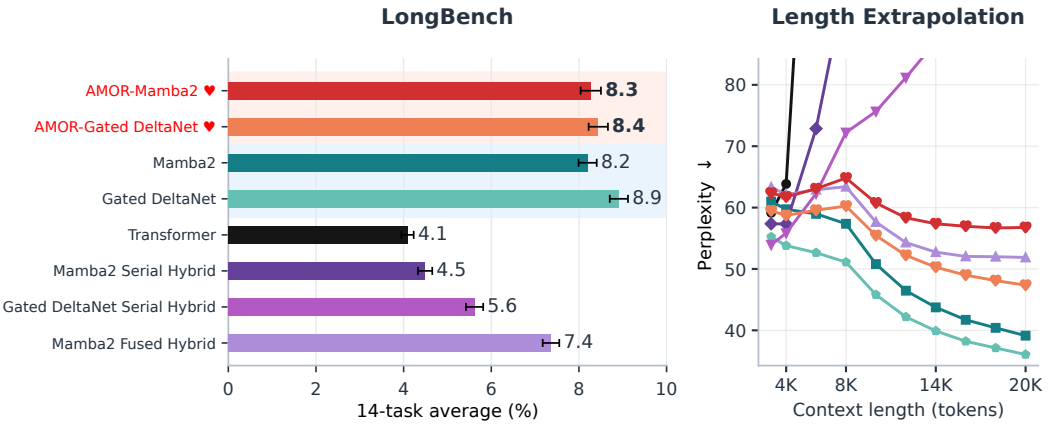

Figure 6: **AMOR retains the long-context performance of recurrent backbones while other attention variants degrade.** Left: 14-task LongBench average. Right: NarrativeQA length extrapolation.

5 Deployment Ablation: Inference Efficiency

To further reduce routing overhead, we optionally distill each block's LM-head entropy estimates into a predictor with a 512-unit hidden layer. The predictors are fitted on the frozen pretrained checkpoint, keeping the pretrained model and gate thresholds fixed. Together, they form a distilled router that adds 3.15M parameters at 1.5B (approximately 0.21%) and replaces the three extra LM-head evaluations (Appendix G). This provides a deployment ablation of the native entropy gate: the pretrained model weights and frozen entropy thresholds are held fixed, while only the mechanism used to estimate

routing entropy is changed. Sections 4.1–4.3 evaluate the native entropy gate; here we evaluate this distilled router.

AMOR-Mamba2 overtakes the fused hybrid in decode between 8K and 16K and the serial hybrid between 32K and 64K (Fig. 7, left). **Its prefill latency is 23.7–40.8% lower than the fused hybrid and tracks closely with the serial hybrid** (middle). **Peak prefill memory remains almost identical to its recurrent backbone's, reaching 29.6% below the fused hybrid and 22.0% below the serial hybrid at 128K** (right). Compared with the native entropy gate, the distilled router lowers peak prefill memory by 2.3–9.2 GB and decode latency by 0.4–1.1 ms/token in the matched 1.5B cohort (Table 11). The post-hoc design runs the recurrent backbone before building attention activations and KV caches, keeping these allocations out of the backbone's memory peak and elegantly redirecting the traffic. At 1.5B, common-sense, retrieval, S-NIAH and LongBench averages change by at most 0.13 points (Table 15); full task results and length extrapolation comparisons appear in Appendix G. The full benchmarking protocol and memory analysis are in Appendix F.3; training and 3K decode throughput appear in Tables 9 and 10. Appendix F.3 reports measurements at all scales: prefill latency and memory improve consistently, while decode gains vary by backbone and context length.

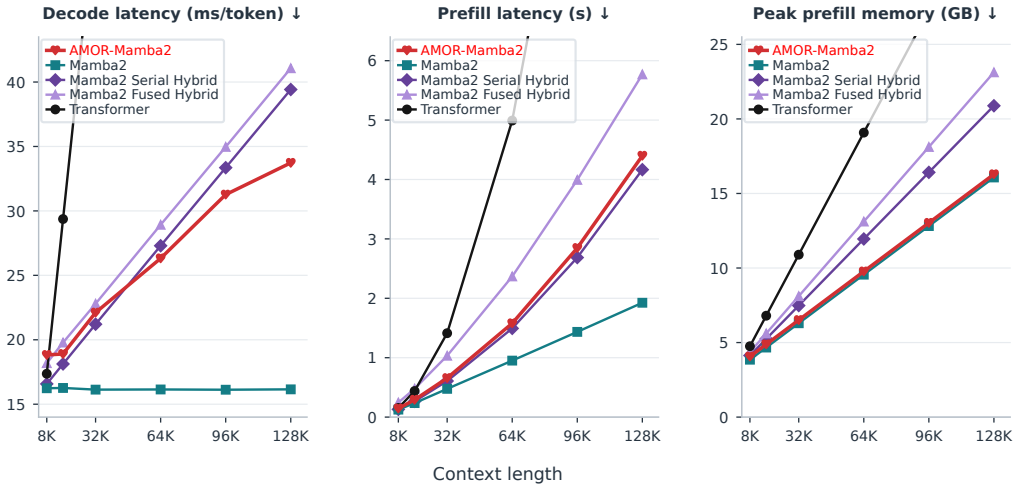

Figure 7: **AMOR exhibits better inference scaling and lower peak prefill memory than its hybrid counterparts.** Left: decode latency. Middle: prefill latency. Right: peak prefill memory. 1.5B; one H100 PCIe, batch 1, memory-efficient SDPA and FineWeb-Edu prompts from 8K to 128K. AMOR uses the distilled router with natural firing. Transformer curves exceed the displayed ranges, and it runs out of memory at 128K. All eight models: Fig. 11; values: Table 11.

6 Conclusion & Future Work

AMOR is a novel hybrid architecture that allocates attention based on the model's predictive uncertainty. Our results suggest that *when* attention is applied matters as much as *how much*: selectively engaging attention based on the model's own uncertainty can improve efficiency and robustness relative to fixed schedules. More broadly, AMOR demonstrates that simple, output-driven mechanisms can effectively coordinate hybrid architectures. Using an entropy gating rule during pretraining and an optional distilled router to reduce deployment overhead, AMOR points toward a class of **models that adapt their computation to the demands of the model.** Future work could learn or calibrate the routing policy itself, potentially adapting the allocation of computation across scales, architectures, and tasks (see limitations in Appendix I).

Ethics Statement. This work studies language-model architectures by pretraining them on existing datasets; the resulting models may inherit biases from their training data.

Reproducibility Statement. The training recipe is provided in Table 6, with architecture, evaluation, baselines, ablations and runtime protocols detailed in the corresponding sections and appendices. The code is provided through an anonymized link in the introduction.

AI Use Statement. Generative AI tools assisted with coding, literature retrieval, and manuscript editing and review. The authors reviewed these contributions and checked all claims against references and executed experimental results. The authors are responsible for the research design and final presentation.

REFERENCES

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language models enable simple systems for generating structured views of heterogeneous data lakes. *Proceedings of the VLDB Endowment*, 17(2):92–105, 2023. URL https://arxiv.org/abs/2304.09433.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In *ICML 2024 Workshop on Efficient Systems for Foundation Models (ES-FoMo)*, 2024a. URL https://arxiv.org/abs/2402.18668.

Simran Arora, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. Just read twice: closing the recall gap for recurrent language models. *arXiv preprint arXiv:2407.05483*, 2024b. URL https://arxiv.org/abs/2407.05483.

Sangmin Bae, Bilge Acun, Chien-Yu Lin, Haroun Habeeb, Seungyeon Kim, Liang Luo, Junjie Wang, and Carole-Jean Wu. Hybrid architectures for language models: Systematic analysis and design insights. *arXiv preprint arXiv:2510.04800*, 2025. URL https://arxiv.org/abs/2510.04800.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL) 2024*, 2024. URL https://arxiv.org/abs/2308.14508.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. In *8th ICML Workshop on Automated Machine Learning 2021*, 2021. URL https://arxiv.org/abs/2107.05407.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024. URL https://arxiv.org/abs/2405.14782.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence 2020*, 2020. URL https://arxiv.org/abs/1911.11641.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS) 2020*, 2020. URL https://arxiv.org/abs/2005.14165.

Linsong Chu, Divya Kumari, Tri Dao, Albert Gu, Raghu Ganti, Dakshi Agrawal, Mudhakar Srivatsa, Davis Wertheimer, Yu Chin Fabian Lim, Antoni Viros, Nelson Gonzalez, Tuan HoangTrong, Ofir Arviv, Yotam Perlitz, Michal Shmueli, Haochen Shen, Minjia Zhang, Gabe Goodhart, Naigang Wang, Nick Hill, Joshua Rosenkranz, Chi-Chun Liu, Adnan Hoque, Chih-Chieh Yang, Sukriti Sharma, Anh Uong, Jay Gala, Syed Zawad, and Ryan Gordon. Bamba: Inference-efficient hybrid mamba2 model. HuggingFace blog, December 2024, 2024. URL https://huggingface.co/blog/bamba.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

CodeParrot. CodeParrot-Clean-Valid. Hugging Face Datasets, 2021. URL https://huggingface.co/datasets/codeparrot/codeparrot-clean-valid.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations (ICLR) 2024*, 2024. URL https://arxiv.org/abs/2307.08691.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML) 2024*, 2024. URL https://arxiv.org/abs/2405.21060.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) 2021*, 2021. URL https://arxiv.org/abs/2105.03011.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024. URL https://arxiv.org/abs/2402.19427.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025. URL https://arxiv.org/abs/2512.02556.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A hybrid-head architecture for small language models. In *International Conference on Learning Representations (ICLR) 2025*, 2025. URL https://arxiv.org/abs/2411.13676.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) 2019*, 2019. URL https://arxiv.org/abs/1903.00161.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL) 2019*, 2019. URL https://arxiv.org/abs/1906.01749.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization (NewSum), co-located with EMNLP-IJCNLP 2019*, 2019. URL https://arxiv.org/abs/1911.12237.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, Anna Golubeva, Vasudev Shyam, James Whittington, Jonathan Pilault, and Beren Millidge. The zamba2 suite: Technical report. *arXiv preprint arXiv:2411.15242*, 2024. URL https://arxiv.org/abs/2411.15242.

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. URL https://arxiv.org/abs/2407.21783.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. URL https://arxiv.org/abs/1603.08983.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Conference on Language Modeling (COLM) 2024*, 2024. URL https://arxiv.org/abs/2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR) 2022*, 2022. URL https://arxiv.org/abs/2111.00396.

Yuxian Gu, Qinghao Hu, Shang Yang, Haocheng Xi, Junyu Chen, Song Han, and Han Cai. Jet-nemotron: Efficient language model with post neural architecture search. *arXiv preprint arXiv:2508.15884*, 2025. URL https://arxiv.org/abs/2508.15884.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian McAuley. Longcoder: A long-range pre-trained language model for code completion. In *International Conference on Machine Learning (ICML) 2023*, 2023. URL https://arxiv.org/abs/2306.14893.

Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: a unified solution for structured web data extraction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*, pp. 775–784, 2011. URL https://dl.acm.org/doi/10.1145/2009916.2010020.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING) 2020*, 2020. URL https://arxiv.org/abs/2011.01060.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training Compute-Optimal Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS) 2022*, 2022. URL https://arxiv.org/abs/2203.15556.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM) 2024*, 2024. URL https://arxiv.org/abs/2404.06654.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) 2021*, 2021. URL https://arxiv.org/abs/2104.02112.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning (ICML) 2024*, 2024. URL https://arxiv.org/abs/2402.01032.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL) 2017 (Volume 1: Long Papers)*, pp. 1601–1611, 2017. URL https://arxiv.org/abs/1705.03551.

Daniel Kahneman. Thinking, fast and slow. Farrar, Straus and Giroux, 2011.

Kimi Team. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025. URL https://arxiv.org/abs/2510.26692.

Kimi Team. Kimi K3: Open Frontier Intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics (TACL)*, 6:317–328, 2018. URL https://arxiv.org/abs/1712.07040.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL)*, 7:453–466, 2019. URL https://aclanthology.org/Q19-1026/.

Aakash Lahoti, Kevin Y. 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 *International Conference on Learning Representations (ICLR) 2026*, 2026. URL https://arxiv.org/abs/2603.15569.

Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING) 2002*, 2002. URL https://aclanthology.org/C02-1150/.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avshalom Manevich, Nir Ratner, Noam Rozen, Erez Schwartz, Mor Zusman, and Yoav Shoham. Jamba: A Hybrid Transformer-Mamba Language Model. *arXiv preprint arXiv:2403.19887*, 2024. URL https://arxiv.org/abs/2403.19887.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Annual Meeting of the Association for Computational Linguistics (ACL) 2022*, 2022. URL https://arxiv.org/abs/2109.07958.

Tianyang Liu, Canwen Xu, and Julian McAuley. Repobench: Benchmarking repository-level code auto-completion systems. In *International Conference on Learning Representations (ICLR) 2024*, 2024. URL https://arxiv.org/abs/2306.03091.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR) 2019*, 2019. URL https://arxiv.org/abs/1711.05101.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*, 2025. URL https://arxiv.org/abs/2502.13189.

William Merrill, Yanhong Li, Tyler Romero, Anej Svete, Caia Costello, Pradeep Dasigi, Dirk Groeneveld, David Heineman, Bailey Kuehl, Nathan Lambert, Chuan Li, Kyle Lo, Saumya Malik, DJ Matusz, Benjamin Minixhofer, Jacob Morrison, Luca Soldaini, Finbarr Timbers, Pete Walsh, Noah A. Smith, Hannaneh Hajishirzi, and Ashish Sabharwal. Olmo hybrid: From theory to practice and back. *arXiv preprint arXiv:2604.03444*, 2026. URL https://arxiv.org/abs/2604.03444.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2018*, 2018. URL https://arxiv.org/abs/1809.02789.

NVIDIA. Nemotron 3 super: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2604.12374*, 2026a. URL https://arxiv.org/abs/2604.12374.

NVIDIA. Nemotron 3 Ultra: Open, Efficient Mixture-of-Experts Hybrid Mamba-Transformer Model for Agentic Reasoning. *arXiv preprint arXiv:2606.15007*, 2026b. URL https://arxiv.org/abs/2606.15007.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Annual Meeting of the Association for Computational Linguistics (ACL) 2016*, 2016. URL https://arxiv.org/abs/1606.06031.

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 *38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks*, 2024. URL https://arxiv.org/abs/2406.17557.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, et al. Rwkv: Reinventing rnns for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://arxiv.org/abs/2305.13048.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient Context Window Extension of Large Language Models. In *International Conference on Learning Representations (ICLR) 2024*, 2024. URL https://arxiv.org/abs/2309.00071.

Qwen Team. Qwen3.5: Towards native multimodal agents. qwen.ai blog / HuggingFace model card, February 2026, 2026. URL https://huggingface.co/Qwen/Qwen3.5-397B-A17B.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR) 2020*, 2020. URL https://arxiv.org/abs/1911.05507.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP) 2016*, pp. 2383–2392, 2016. URL https://arxiv.org/abs/1606.05250.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024. URL https://arxiv.org/abs/2404.02258.

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 *International Conference on Learning Representations (ICLR) 2025*, 2025. URL https://arxiv.org/abs/2406.07522.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence 2020*, 2020. URL https://arxiv.org/abs/1907.10641.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems (NeurIPS) 2022 (Oral)*, 2022. URL https://arxiv.org/abs/2207.07061.

Noam Shazeer. GLU Variants Improve Transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

Kate Soule and Dave Bergmann. IBM Granite 4.0: Hyper-efficient, High Performance Hybrid Models for Enterprise. IBM announcement, October 2025, 2025. URL https://www.ibm.com/new/announcements/ibm-granite-4-0-hyper-efficient-high-performance-hybrid-models.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics (TACL)*, 10:539–554, 2022. URL https://arxiv.org/abs/2108.00573.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS) 30*, 2017. URL https://arxiv.org/abs/1706.03762.

Dustin Wang, Rui-Jie Zhu, Steven Abreu, Yong Shan, Taylor Kergan, Yuqi Pan, Yuhong Chou, Zheng Li, Jibin Wu, Ge Zhang, Wenhao Huang, and Jason Eshraghian. A Systematic Analysis of Hybrid Linear Attention. *arXiv preprint arXiv:2507.06457*, 2025. URL https://arxiv.org/abs/2507.06457.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems (NeurIPS) 2024*, 2024. URL https://arxiv.org/abs/2406.06484.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations (ICLR) 2025*, 2025. URL https://arxiv.org/abs/2412.06464.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2018*, 2018. URL https://arxiv.org/abs/1809.09600.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025. URL https://arxiv.org/abs/2502.11089.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL) 2019*, 2019. URL https://arxiv.org/abs/1905.07830.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, 2019. URL https://arxiv.org/abs/1910.07467.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) 2021*, 2021. URL https://arxiv.org/abs/2104.05938.

Jingwei Zuo, Maksim Velikanov, Ilyas Chahed, Younes Belkada, Dhia Eddine Rhayem, Guillaume Kunsch, Hakim Hacid, Hamza Yous, Brahim Farhat, Ibrahim Khadraoui, Mugariya Farooq, Giulia Campesan, Ruxandra Cojocaru, Yasser Djilali, Shi Hu, Iheb Chaabane, Puneesh Khanna, Mohamed El Amine Seddik, Ngoc Dung Huynh, Phuc Le Khac, Leen AlQadi, Billel Mokeddem, Mohamed Chami, Abdalgader Abubaker, Mikhail Lubinets, Kacper Piskorski, and Slim Frikha. Falcon-h1: A family of hybrid-head language models redefining efficiency and performance. *arXiv preprint arXiv:2507.22448*, 2025. URL https://arxiv.org/abs/2507.22448.

A Full Benchmark Tables

Tables 1–3 give the full results underlying the main-text figures. Common-sense results include repeated-evaluation SDs, except training-time FineWeb-Edu perplexity (FW-Edu PPL); LongBench includes benchmark standard errors. LM Evaluation Harness returned N/A for retrieval and S-NIAH standard errors in our evaluations, so we report point estimates.

Table 1: **Common-sense results across scales.** Scores are percentages unless marked PPL; ± denotes the sample SD of five reevaluations per checkpoint.

Model	FW-Edu PPL ↓	LAMB PPL ↓	LAMB acc ↑	HSwag acc ↑	PIQA acc ↑	ARC-E acc ↑	ARC-C acc ↑	WinoG acc ↑	OBQA acc ↑	TQA mc2 acc ↑	Avg acc ↑
180M											
Transformer	31.96	402.56±0.00	18.3±0.00	27.2±0.00	60.2±0.00	47.6±0.00	18.4±0.00	50.3±0.00	15.6±0.00	45.0±0.00	35.3±0.00
Mamba2	32.47	352.28±0.02	16.8±0.00	27.4±0.01	60.0±0.12	45.7±0.09	18.5±0.00	48.9±0.13	**17.2**±0.00	43.8±0.02	34.8±0.02
Gated DeltaNet	31.17	392.58±0.08	15.6±0.04	27.5±0.01	60.9±0.10	47.4±0.04	19.0±0.04	52.1±0.11	14.2±0.00	43.6±0.02	35.0±0.01
Mamba2 Serial Hybrid	30.69	398.90±0.01	18.8±0.01	27.5±0.01	60.6±0.07	46.8±0.04	**19.7**±0.04	**53.0**±0.11	15.4±0.09	43.0±0.01	35.6±0.03
Gated DeltaNet Serial Hybrid	**30.24**	366.69±0.02	18.1±0.03	27.3±0.01	**61.6**±0.00	45.7±0.02	18.6±0.00	52.6±0.14	15.8±0.00	41.3±0.01	35.1±0.02
Mamba2 Fused Hybrid	30.73	304.32±0.09	18.7±0.01	**27.6**±0.00	60.1±0.00	45.5±0.07	18.6±0.05	49.4±0.09	16.0±0.00	44.0±0.04	35.0±0.00
AMOR-Mamba2 ♥	31.46	240.27±0.04	**20.3**±0.06	27.4±0.00	60.5±0.12	46.5±0.05	18.0±0.09	51.3±0.22	16.6±0.00	**45.9**±0.03	**35.8**±0.04
AMOR-Gated DeltaNet ♥	30.60	**234.20**±0.02	19.5±0.05	27.3±0.01	61.0±0.04	**47.7**±0.05	17.3±0.00	51.1±0.11	16.2±0.00	45.2±0.01	35.7±0.01
440M											
Transformer	19.92	74.09±0.00	27.8±0.00	31.2±0.00	64.9±0.00	55.9±0.00	23.9±0.00	49.9±0.00	20.4±0.00	**42.4**±0.00	39.5±0.00
Mamba2	19.78	74.19±0.00	25.3±0.00	31.9±0.00	65.2±0.00	**56.9**±0.00	23.2±0.00	51.1±0.00	21.6±0.00	36.7±0.03	39.0±0.00
Gated DeltaNet	19.39	62.37±0.01	26.6±0.02	31.9±0.02	**66.3**±0.05	56.7±0.06	23.4±0.08	50.5±0.21	21.2±0.09	40.7±0.03	39.7±0.02
Mamba2 Serial Hybrid	19.11	69.31±0.01	28.6±0.03	**32.0**±0.01	66.1±0.02	55.9±0.02	23.6±0.11	50.8±0.18	20.4±0.00	38.8±0.01	39.5±0.03
Gated DeltaNet Serial Hybrid	**18.94**	56.21±0.00	29.6±0.01	31.8±0.02	65.8±0.00	**56.9**±0.07	24.3±0.14	52.0±0.09	21.6±0.11	39.5±0.03	40.2±0.01
Mamba2 Fused Hybrid	19.24	58.17±0.00	28.9±0.04	31.9±0.00	65.8±0.00	56.1±0.02	24.3±0.19	51.9±0.18	22.2±0.00	**42.4**±0.02	40.4±0.04
AMOR-Mamba2 ♥	19.60	57.37±0.02	29.3±0.00	31.4±0.03	65.9±0.12	55.4±0.16	**24.7**±0.05	**52.4**±0.26	21.0±0.00	36.4±0.01	39.6±0.06
AMOR-Gated DeltaNet ♥	19.27	**50.15**±0.00	**30.2**±0.04	31.8±0.02	65.0±0.05	56.0±0.06	24.6±0.15	51.3±0.18	**23.8**±0.00	40.9±0.01	**40.5**±0.04
1.5B											
Transformer	13.41	25.93±0.00	38.2±0.00	38.3±0.00	69.5±0.00	65.2±0.00	29.9±0.00	52.2±0.00	23.8±0.00	35.4±0.00	44.1±0.00
Mamba2	13.32	26.75±0.00	36.1±0.01	38.9±0.02	70.3±0.12	66.1±0.07	30.4±0.05	**55.1**±0.52	25.6±0.00	34.3±0.00	44.6±0.03
Gated DeltaNet	13.11	22.03±0.00	38.5±0.08	39.2±0.02	70.7±0.05	67.7±0.02	**33.4**±0.09	53.1±0.25	25.0±0.00	35.0±0.00	45.3±0.06
Mamba2 Serial Hybrid	12.97	25.28±0.00	38.1±0.00	**39.4**±0.00	**70.9**±0.00	**67.9**±0.00	31.3±0.00	53.0±0.00	26.0±0.00	35.4±0.02	45.3±0.00
Gated DeltaNet Serial Hybrid	**12.94**	22.28±0.00	**39.7**±0.05	39.2±0.03	70.8±0.03	64.9±0.00	30.5±0.00	54.2±0.00	24.6±0.00	37.7±0.02	45.2±0.01
Mamba2 Fused Hybrid	13.04	23.71±0.00	37.5±0.05	39.2±0.00	70.4±0.12	67.4±0.05	32.3±0.09	53.7±0.13	26.6±0.00	35.2±0.00	45.3±0.02
AMOR-Mamba2 ♥	13.30	23.01±0.01	39.2±0.02	38.6±0.01	**70.9**±0.05	67.0±0.05	30.2±0.07	54.7±0.16	**26.8**±0.00	**39.0**±0.02	**45.8**±0.03
AMOR-Gated DeltaNet ♥	13.10	**21.49**±0.01	39.1±0.06	39.2±0.02	70.6±0.07	66.4±0.02	30.5±0.00	53.7±0.07	24.8±0.09	36.9±0.00	45.1±0.02

Table 2: **Retrieval and S-NIAH.** 1.5B; retrieval measures answer-containment accuracy at 2K; S-NIAH reports accuracy by variant and context length. Scores are percentages.

Model	SWDE↑	SQuAD↑	FDA↑	TQA↑	NQ↑	DROP↑	S-NIAH-1↑			S-NIAH-2↑			S-NIAH-3↑		
Context length	2048						1024	2048	4096	1024	2048	4096	1024	2048	4096
Transformer	54.5	22.7	40.1	41.7	10.5	18.3	**100.0**	**100.0**	12.6	**100.0**	**100.0**	14.6	90.6	87.4	0.2
Mamba2	19.9	33.5	14.5	39.6	8.7	17.0	**100.0**	93.6	60.4	**100.0**	38.4	28.4	72.4	40.2	14.4
Gated DeltaNet	25.3	35.4	11.8	38.7	9.6	18.1	99.4	91.4	60.8	99.4	17.8	38.8	71.6	47.2	24.2
Mamba2 Serial Hybrid	49.1	**44.6**	37.9	**43.1**	**10.8**	18.4	**100.0**	99.6	68.6	**100.0**	**100.0**	64.0	91.2	**91.0**	20.4
Gated DeltaNet Serial Hybrid	**57.1**	41.5	**49.4**	**43.1**	10.7	19.8	**100.0**	**100.0**	67.8	**100.0**	**100.0**	34.0	**91.4**	90.8	28.4
Mamba2 Fused Hybrid	45.1	39.9	48.5	38.4	9.6	18.1	**100.0**	**100.0**	**81.4**	**100.0**	**100.0**	**69.6**	89.6	90.2	22.8
AMOR-Mamba2 ♥	33.8	36.0	21.2	40.4	10.2	**21.1**	**100.0**	95.8	44.4	**100.0**	99.6	47.2	70.8	83.6	25.6
AMOR-Gated DeltaNet ♥	47.4	36.8	24.9	**43.1**	10.5	17.4	**100.0**	96.2	48.0	**100.0**	98.6	53.0	88.8	86.0	**29.4**

Table 3: **LongBench.** 1.5B; scores are percentages; ± denotes standard error.

Model	Single-Doc QA↑			Multi-Doc QA↑			Summarization↑			Few-shot↑			Code↑		Avg↑
	NQA	QQA	MFQ	HQA	2WM	Mus	GvR	QMS	MNs	TRC	TQA	SSM	LCC	RBP	
Transformer	0.3±0.11	1.6±0.18	4.9±0.65	0.6±0.31	1.4±0.38	0.2±0.13	5.3±0.24	5.8±0.37	10.7±0.49	3.0±1.21	3.3±0.93	3.6±0.49	9.0±0.27	7.7±0.31	4.1±0.14
Mamba2	1.6±0.16	3.9±0.40	11.1±0.91	**4.6**±0.60	6.8±0.74	**2.8**±0.44	7.5±0.28	**15.6**±0.58	11.3±0.43	7.0±1.81	14.3±1.33	7.2±0.57	10.1±0.33	11.0±0.44	8.2±0.21
Gated DeltaNet	1.4±0.15	3.6±0.33	9.4±0.96	3.3±0.44	**8.3**±0.80	2.4±0.37	8.5±0.27	14.9±0.56	10.8±0.40	5.5±1.62	**17.0**±1.48	**19.3**±0.93	9.5±0.29	10.8±0.40	**8.9**±0.21
Mamba2 Serial Hybrid	0.2±0.07	2.2±0.31	4.4±0.79	0.2±0.10	2.5±0.49	0.4±0.17	2.8±0.25	4.5±0.30	10.0±0.53	4.5±1.47	4.2±1.05	6.6±0.57	10.0±0.29	10.3±0.32	4.5±0.16
Gated DeltaNet Serial Hybrid	0.5±0.10	2.4±0.26	5.6±0.70	1.3±0.27	4.8±0.67	1.1±0.26	4.1±0.28	5.6±0.39	10.5±0.47	7.5±1.87	9.2±1.34	5.4±0.68	9.6±0.33	11.2±0.36	5.6±0.20
Mamba2 Fused Hybrid	1.2±0.14	3.4±0.34	9.5±0.82	3.1±0.47	6.4±0.80	2.2±0.40	**9.3**±0.29	14.8±0.50	8.1±0.40	3.0±1.21	14.2±1.57	6.0±0.52	**11.2**±0.53	10.8±0.47	7.4±0.19
AMOR-Mamba2 ♥	1.6±0.15	3.3±0.32	10.1±0.83	2.9±0.50	6.1±0.71	1.8±0.37	7.6±0.23	14.7±0.46	**13.2**±0.47	**13.0**±2.38	13.2±1.49	7.5±0.51	9.3±0.26	**11.5**±0.29	8.3±0.23
AMOR-Gated DeltaNet ♥	**1.7**±0.17	**4.0**±0.35	**11.4**±0.87	3.6±0.49	7.0±0.74	1.9±0.32	8.1±0.30	15.1±0.50	10.8±0.43	11.0±2.22	11.8±1.22	11.4±0.68	10.1±0.38	10.3±0.41	8.4±0.22

LongBench standard errors come from the stored harness outputs. The average's standard error is $\sqrt{\sum_i \mathrm{SE}_i^2}/14$, assuming independent tasks; it measures fixed-checkpoint sampling error.

A.1 LENGTH EXTRAPOLATION RESULTS

RoPE-induced distribution shift beyond the 3K training context length leads to sharp perplexity increases for the Transformer and the two serial hybrids (Fig. 8). AMOR does not suffer as much, behaving more similarly to recurrent models. For visualization clarity, the y-axes in each panel are linearly capped at $1.3\times$ the maximum value observed among the well-behaved subset $\{$Mamba2, Gated DeltaNet, Mamba2 Fused Hybrid, AMOR-Mamba2, AMOR-Gated DeltaNet$\}$, ensuring that extreme outliers do not dominate the scale while preserving relative trends among stable models.

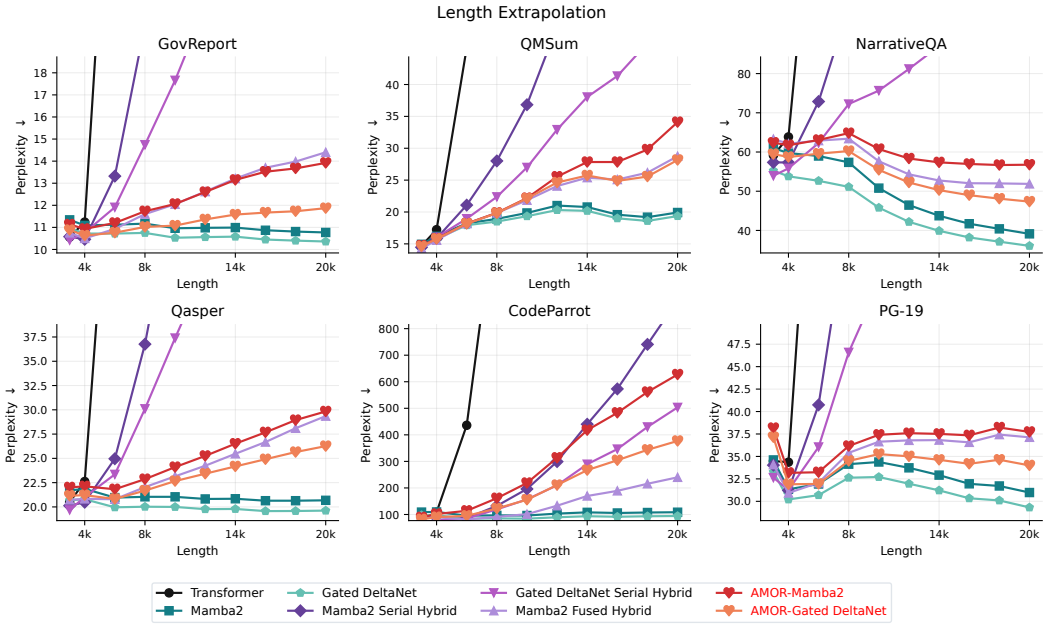

Figure 8: **Length extrapolation across six domains.** 1.5B; per-token perplexity from the 3K training context to 20K; lower is better.

A.2 REPEATABILITY AND TRAINING-SEED RESULTS

Evaluation repeatability. We reevaluate each fixed checkpoint five times. Table 1 reports canonical scores with sample SDs from these repeats; the average's SD is computed across the five eight-task means. The largest SD is 0.06 points for an eight-task mean and 0.52 points for a single task (WinoGrande).

Independent training runs. At 180M, every architecture is additionally trained with seeds 37, 67, 69 and 78, each trained to a Chinchilla-optimal token budget. Sample SDs of the eight-task mean range from 0.20 to 0.48 points (median 0.24; Table 4). Seeds 37, 67, 69 and 78 control initialization and training-data order. The canonical seed-42 run is excluded from these statistics. Consistent with the canonical seed-42 results in the main paper, an AMOR variant leads at each training seed and in the average across all four seeds.

Table 4: **Eight-task common-sense averages at 180M across training seeds.** Evaluation settings follow Appendix E.1.

Model	Common-sense avg (acc, %) ↑					
	Seed 37	Seed 67	Seed 69	Seed 78	Avg	SD
Transformer	34.6	35.1	35.1	34.9	34.9	0.24
Mamba2	35.1	35.0	34.7	34.7	34.9	0.20
Gated DeltaNet	35.4	35.6	34.8	35.0	35.2	0.40
Mamba2 Serial Hybrid	35.5	35.2	35.5	35.1	35.3	0.23
Gated DeltaNet Serial Hybrid	36.0	35.5	34.9	35.1	35.4	0.48
Mamba2 Fused Hybrid	35.4	35.3	35.6	35.1	35.3	0.21
AMOR-Mamba2 ♥	**36.2**	35.5	35.7	**35.8**	**35.8**	0.28
AMOR-Gated DeltaNet ♥	35.6	**35.9**	**36.0**	35.6	**35.8**	0.20

B ARCHITECTURAL DETAILS

All eight architectures share model width d_{model} and backbone depth N. The reference dimensions are $d_{\text{model}}/N/d_{\text{ff}} = 768/12/1216$ (180M), $1024/24/1984$ (440M), and $2048/24/4096$ (1.5B), where d_{ff} is the SwiGLU hidden width (Shazeer, 2020). Table 5 reports parameter counts and attention placement. Canonical AMOR appends $K = 3$ blocks to the N mixer–MLP layers; the $K = 1$ ablation covers both backbones and all scales (Table 29).

Controlled design. To isolate the effect of *fixed-schedule attention*, we remove orthogonal components from the canonical Jamba and Hymba designs. All hybrids use full causal attention rather than sliding-window attention, and use the same corresponding Mamba2 or Gated DeltaNet backbone. We exclude Hymba's meta tokens and KV reuse, as well as Jamba's mixture-of-experts layers.

Fixing backbone depth does not equalize parameter counts because the sequence mixers have different parameterizations. The additional AMOR blocks introduce a modest parameter overhead relative to their recurrent backbones (approximately 2–4%; Table 5). The Transformer has the fewest parameters at each scale because its attention mixers are lighter than the recurrent mixers in this comparison.

Table 5: **Architecture configurations across scales.** 180M, 440M and 1.5B; parameters, attention placement, head dimension and backbone depth. N/A denotes no attention layers.

Model	Params			Attn layers	Attn dim	N
	180M	440M	1.5B			
Transformer	160.5M	378.3M	1,269.4M	all N	64 head	12 / 24 / 24
Mamba2	177.3M	436.1M	1,487.1M	0	N/A	12 / 24 / 24
Gated DeltaNet	174.9M	429.5M	1,473.7M	0	N/A	12 / 24 / 24
Mamba2 Serial Hybrid	174.5M	428.8M	1,459.9M	$\{4, 8\}$ / $\{6, 12, 18\}$	64 head	12 / 24 / 24
Gated DeltaNet Serial Hybrid	172.5M	423.1M	1,448.1M	$\{4, 8\}$ / $\{6, 12, 18\}$	64 head	12 / 24 / 24
Mamba2 Fused Hybrid	179.1M	439.2M	1,499.7M	$\{0, 6, 11\}$ / $\{0, 12, 23\}$	64 head	12 / 24 / 24
AMOR-Mamba2 ♥	184.4M	448.7M	1,537.4M	K=3 post-hoc	64 head	12 / 24 / 24
AMOR-Gated DeltaNet ♥	182.0M	442.1M	1,524.0M	K=3 post-hoc	64 head	12 / 24 / 24

Block normalization and the AMOR block design are described in Section 3.1; symmetric-layout architecture ablation results are in Appendix H.

C TRAINING RECIPE

Canonical runs share the training corpus, Llama-3.1 tokenizer (Grattafiori et al., 2024) and AdamW (Loshchilov & Hutter, 2019) recipe within each scale (Table 6); seed 42 controls initialization and training-data order. The common token budget is 20 times the parameter count of AMOR-Mamba2, following Chinchilla scaling (Hoffmann et al., 2022) and giving every architecture the same data volume at that scale. Results across training seeds are in Appendix A.2.

Table 6: **All architectures share one training recipe per scale.** 180M, 440M and 1.5B; AMOR's $\boldsymbol{\alpha}$ uses a separate AdamW group without weight decay.

	180M	440M	1.5B
Tokens	3.69B	8.97B	30.7B
Steps	7,503	18,255	62,558
Sequence length	3,072	3,072	3,072
Effective batch (tokens)	491,520	491,520	491,520
Optimizer	AdamW (β_1, β_2)=(0.9, 0.95), weight decay 0.1, grad clip 1.0		
Backbone peak LR	3×10^{-4}	3×10^{-4}	1.5×10^{-4}
LR floor	10^{-5}	10^{-5}	10^{-5}
LR warmup steps	2,000	2,000	2,000
LR schedule	Cosine		
Precision	bfloat16 autocast		
Tokenizer	Llama-3.1, V=128,256		
Seed	42	42	42
AMOR-*specific (separate AdamW group)*			
$\boldsymbol{\alpha}$ peak LR	3×10^{-3}	3×10^{-3}	3×10^{-3}
$\boldsymbol{\alpha}$ weight decay	0	0	0
Init $\tilde{\boldsymbol{\alpha}}$	0 ($\boldsymbol{\alpha}$=0.5)	0	0
Init $\mathbf{W}_O$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
EMA momentum	0.99	0.99	0.99
κ (offset multiplier)	0.2	0.2	0.2

After construction, we initialize embedding and linear projection weights from $\mathcal{N}(0, 0.02)$, except the AMOR attention output projections, which start at zero. Normalization weights and recurrent-specific parameters retain their respective initializations. The embedding and LM head are tied, and the canonical models' linear projections have no bias.

C.1 TRAINING MODE

We train in $\mathrm{full_with_mask}$ mode, in which $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ and the dense causal SDPA are computed for every position, and the gate masks the attention output per position. This is mathematically equivalent to $\mathrm{true_sparse}$ formulation that computes attention only at firing positions; both modes produce the same residual update in exact arithmetic.

We use the dense formulation for training purely for efficiency: it maps to PyTorch's `scaled_dot_product_attention` (FlashAttention-2 kernels), whereas $\mathrm{true_sparse}$ requires per-row dynamic gather/scatter without a comparable fused implementation, resulting in lower GPU throughput. The choice therefore affects only implementation efficiency, not model semantics.

D ENTROPY GATE BEHAVIOR

D.1 TRAINING STATISTICS

Table 7 summarizes Block 0 training statistics over the final logged entropy window: mean batch median $\bar{\mu}$, mean batch SD $\bar{\sigma}$, derived threshold $\bar{\tau} = \bar{\mu} + 0.2\bar{\sigma}$, and the mean per-channel scaling vector $\boldsymbol{\alpha}$. The derived threshold is not a direct readout of the frozen EMA buffers at the final checkpoint. With $\kappa = 0.2$ fixed, entropy medians rise and their standard deviations fall with scale, while the 10th–90th percentile ranges of logged batch firing rates remain near 33–51%. Figure 9 shows per-block entropy histograms and frozen entropy thresholds at 1.5B. Each histogram aggregates the final 20 entropy-log entries for its block, sampled every 500 steps from step 53,000 to 62,500 (approximately 15% of training). Firing-rate annotations average the logged rates over that window.

Table 7: **Training firing-rate ranges remain similar across scales.** 180M, 440M and 1.5B; Block 0 statistics: $\bar{\mu}$ and $\bar{\sigma}$ are mean batch median and SD, $\bar{\tau} = \bar{\mu} + 0.2\bar{\sigma}$.

Model	Scale	FW-Edu PPL ↓	$\bar{\mu}$	$\bar{\sigma}$	$\bar{\tau}$	α mean	Training firing rate 10th–90th percentile
			normalized entropy				
AMOR-Mamba2 ♥	180M	31.46	0.706	0.106	0.728	0.801	33.1–45.8%
AMOR-Mamba2 ♥	440M	19.60	0.847	0.057	0.858	0.810	34.1–47.6%
AMOR-Mamba2 ♥	1.5B	13.30	0.934	0.028	0.940	0.662	32.9–49.4%
AMOR-Gated DeltaNet ♥	180M	30.60	0.707	0.108	0.728	0.779	33.3–45.5%
AMOR-Gated DeltaNet ♥	440M	19.27	0.853	0.053	0.864	0.613	33.9–46.9%
AMOR-Gated DeltaNet ♥	1.5B	13.10	0.937	0.030	0.943	0.544	32.6–50.6%

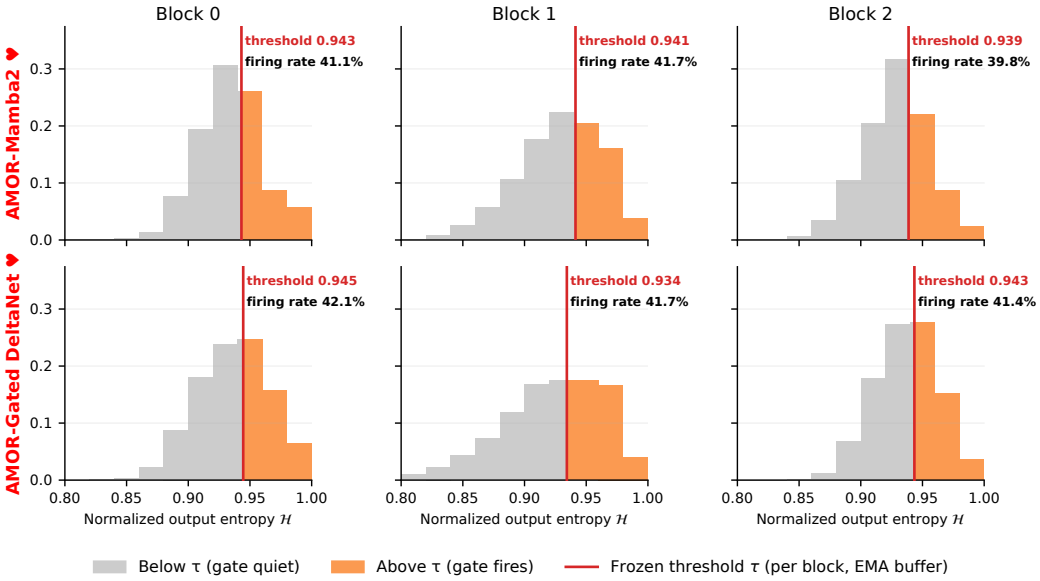

Figure 9: **Training entropy concentrates near the frozen thresholds.** 1.5B; final-window histograms. Red lines mark frozen entropy thresholds, and orange marks mass above them.

D.2 Held-out gating behavior

For AMOR variants pretrained with a target firing rate of $\sim$40% at 1.5B, observed firing rates during inference prefill on held-out text vary by domain but remain similar from 3K to 20K within each dataset (training context: 3K). Rates are 37–43% on FineWeb-Edu (Penedo et al., 2024), 64–69% on PG-19 (Rae et al., 2020), and 97–99% on CodeParrot (CodeParrot, 2021) (Table 8). During generative decoding from a 3K held-out FineWeb-Edu prompt, firing rates range from 18–31% across greedy and sampled decoding with the native entropy gate or distilled router (Table 10).

Table 8: **Prefill firing varies more by domain than context length.** 1.5B; firing rates average all positions and three blocks over six text segments per context length.

Prefill firing rate on held-out text					
Model	**Dataset**	3K	4K	8K	20K
AMOR-Mamba2 ♥	FineWeb-Edu	38.0%	37.6%	38.0%	37.2%
	PG-19	64.1%	66.4%	67.4%	64.9%
	CodeParrot	98.9%	98.9%	98.3%	97.1%
AMOR-Gated DeltaNet ♥	FineWeb-Edu	42.7%	41.8%	40.5%	40.8%
	PG-19	66.9%	69.1%	68.1%	64.4%
	CodeParrot	99.4%	99.4%	99.0%	98.2%

D.3 Additional gate-fire examples

Figure 10 shows four selected prompts. Individual blocks fire on 27–74% of the displayed tokens, including rare subword pieces, numbers and names, while remaining closed on locally predictable continuations.

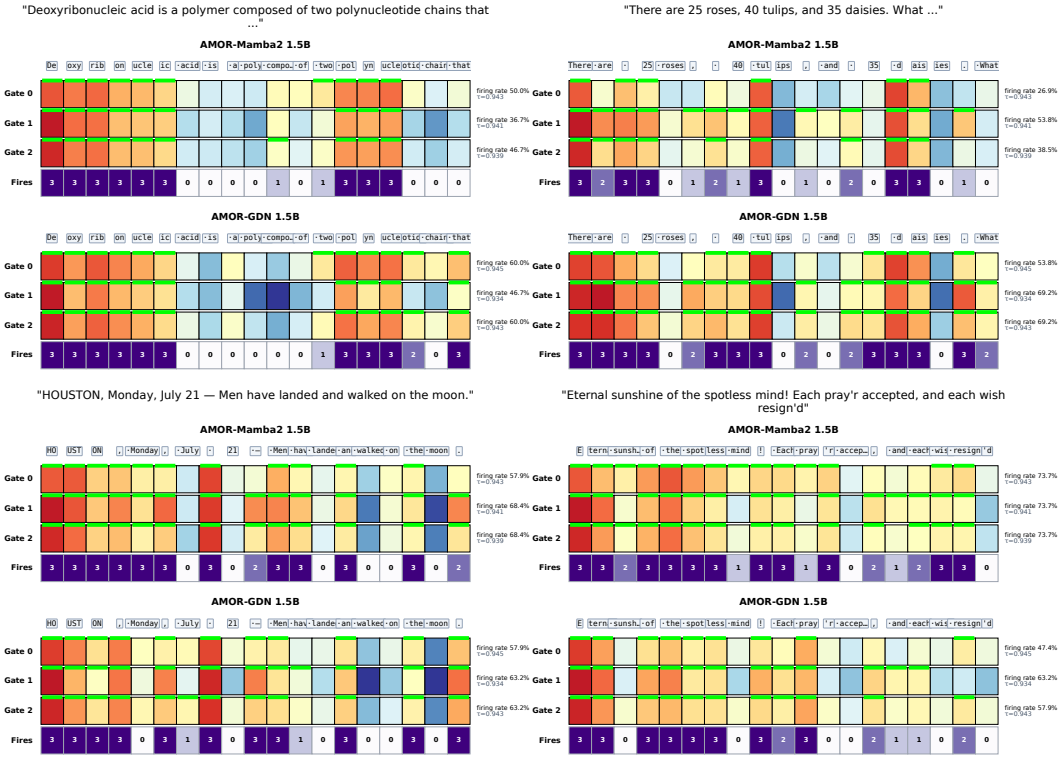

Figure 10: **Gate patterns vary across domains.** 1.5B; native entropy gates for AMOR-Mamba2 and AMOR-Gated DeltaNet on DNA, mathematical reasoning, news and poetry prompts. Colors follow Fig. 1.

E Evaluation Setup

Downstream benchmark evaluations run via the LM Evaluation Harness (Biderman et al., 2024); loglikelihood tasks use the harness defaults, and generation tasks use greedy decoding.

E.1 COMMON-SENSE REASONING

We evaluate all models on eight common-sense zero-shot loglikelihood tasks across 180M, 440M, and 1.5B scales. The tasks are: LAMBADA (Paperno et al., 2016) (last-word prediction in narrative passages), HellaSwag (Zellers et al., 2019) (multiple-choice sentence completion stressing physical and temporal commonsense), PIQA (Bisk et al., 2020) (physical-interaction QA), ARC-Easy and ARC-Challenge (Clark et al., 2018) (grade-school science multiple-choice), WinoGrande (Sakaguchi et al., 2020) (pronoun-disambiguation Winograd schemas), OpenBookQA (Mihaylov et al., 2018) (open-book elementary-science QA), and TruthfulQA-mc2 (Lin et al., 2022) (multiple-choice questions whose distractors are popular misconceptions).

Every common-sense accuracy score in this paper, in Tables 1, 4, 26, 27, 29 and 30 and in Fig. 4, is the harness's raw accuracy (`acc`).

E.2 IN-CONTEXT RETRIEVAL

Following Gated DeltaNet and Mamba3 (Yang et al., 2025; Lahoti et al., 2026), we evaluate all eight models at 1.5B for retrieval, varying context length.

Retrieval evaluation comprises two task suites. The first uses the six-task cloze-completion protocol of Arora et al. (2024b), truncated to a 2K context window. It includes SWDE (Hao et al., 2011) for structured HTML relation extraction, FDA (Arora et al., 2023) for PDF key–value retrieval, SQuAD-completion (Rajpurkar et al., 2016) for short-passage extractive question answering, and cloze-formatted variants of TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and DROP (Dua et al., 2019). The second suite is Single Needle-in-a-Haystack (S-NIAH), drawn from RULER (Hsieh et al., 2024) and evaluated at context lengths of 1K, 2K, and 4K. It consists of S-NIAH-1 (passkey retrieval in a repeated synthetic haystack), S-NIAH-2 (numeric needle embedded in a real-essay haystack), and S-NIAH-3 (UUID needle embedded in a real-essay haystack).

E.3 LONG-CONTEXT BEHAVIOR

Long-context comparisons follow the same convention as Gated DeltaNet and Mamba3 (Yang et al., 2025; Lahoti et al., 2026), using 1.5B models across varying context lengths.

E.3.1 LONGBENCH

We evaluate 14 tasks from **LongBench** (Bai et al., 2024): single-document QA on NarrativeQA (Kočiský et al., 2018), Qasper (Dasigi et al., 2021), and MultiFieldQA-en; multi-document QA on HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022); summarization on GovReport (Huang et al., 2021), QMSum (Zhong et al., 2021), and MultiNews (Fabbri et al., 2019); few-shot in-context learning on TREC (Li & Roth, 2002), TriviaQA (Joshi et al., 2017), and SAMSum (Gliwa et al., 2019); and code completion on LCC (Guo et al., 2023) and RepoBench-P (Liu et al., 2024).

E.3.2 LENGTH EXTRAPOLATION

We set out to explore per-token perplexity as a function of context length, to assess the model's length extrapolation capability beyond its training distribution (see Fig. 8). We computed the per-token perplexity at ten lengths ranging from 3K (the training context) to 20K tokens, using 50 segments per length per dataset. We follow the six benchmarks used in Yang et al. (2025) for length extrapolation: GovReport (Huang et al., 2021) (long-form government reports), QMSum (Zhong et al., 2021) (query-based meeting summarization), NarrativeQA (Kočiský et al., 2018) (literary-passage QA), Qasper (Dasigi et al., 2021) (research-paper QA), CodeParrot (CodeParrot, 2021) (GitHub-Python code), and PG-19 (Rae et al., 2020) (long-form public-domain books).

F EFFICIENCY MEASUREMENTS

F.1 TRAINING THROUGHPUT

AMOR reaches 85–87% of its recurrent backbone's throughput; AMOR-Mamba2 is 1.4× faster than the Mamba2 fused hybrid. Forward FLOPs include each AMOR block's LM-head evaluation.

Table 9: **AMOR retains most recurrent-model training throughput.** 1.5B; one H100 NVL, 100 training steps with microbatch 2 and 80 accumulation steps.

Model	**Fwd FLOPs** GFLOPs/tok ↓	**Throughput** thousand tok/s ↑	**Peak training memory** GiB ↓
Transformer	3.15	23.59	44.35
Mamba2	2.60	25.09	48.07
Gated DeltaNet	2.95	22.82	53.60
Mamba2 Serial Hybrid	2.67	24.61	47.61
Gated DeltaNet Serial Hybrid	2.97	23.36	52.45
Mamba2 Fused Hybrid	2.70	15.31	49.52
AMOR-Mamba2 ♥	4.35	21.46	49.63
AMOR-Gated DeltaNet ♥	4.70	19.83	55.16

F.2 SHORT-CONTEXT DECODING THROUGHPUT

With batch size 1, each trial prefills the same 3,072-token prompt from held-out FineWeb-Edu text, then generates 256 tokens. We report throughput from the mean decoding time over three trials, following a separate 32-token warm-up.

Table 10: **At 3K, AMOR tracks recurrent-model decode throughput.** 1.5B; one H100 NVL. Firing rates average generated positions and three blocks. T denotes sampling temperature; N/A denotes no gate.

Model	**Greedy**, T=0		**Sampling**, T=1	
	tok/s ↑	firing rate	tok/s ↑	firing rate
Transformer	102.0	N/A	100.5	N/A
Mamba2	63.8	N/A	64.6	N/A
Gated DeltaNet	44.6	N/A	44.4	N/A
Mamba2 Serial Hybrid	64.5	N/A	64.7	N/A
Gated DeltaNet Serial Hybrid	48.6	N/A	48.1	N/A
Mamba2 Fused Hybrid	60.1	N/A	59.8	N/A
AMOR-Mamba2 ♥	57.8	20.8%	57.9	24.2%
+ distilled router	58.3	20.6%	58.0	17.6%
AMOR-Gated DeltaNet ♥	41.7	30.1%	41.9	19.4%
+ distilled router	41.9	30.7%	41.4	20.6%

F.3 LONG-CONTEXT INFERENCE EFFICIENCY

Table 11: **Inference efficiency.** 1.5B; one H100 PCIe, batch 1, memory-efficient SDPA. Prefill latency and peak prefill memory use one warmed measurement with final-position logits. Decode latency averages three 32-token runs after eight warm-up tokens. OOM denotes out of memory.

Model	Context length					
	8K	16K	32K	64K	96K	128K
Prefill latency (s) ↓						
Transformer	0.15	0.44	1.41	4.99	10.83	OOM
Mamba2	0.13	0.24	0.48	0.95	1.43	1.92
Gated DeltaNet	0.11	0.20	0.40	0.80	1.21	1.61
Mamba2 Serial Hybrid	0.13	0.27	0.61	1.49	2.69	4.17
Gated DeltaNet Serial Hybrid	0.11	0.24	0.54	1.37	2.49	3.89
Mamba2 Fused Hybrid	0.24	0.48	1.03	2.37	3.99	5.77
AMOR-Mamba2 ♥	0.26	0.53	1.13	2.53	4.27	6.30
+ distilled router	0.14	0.29	0.66	1.58	2.85	4.40
AMOR-Gated DeltaNet ♥	0.24	0.49	1.05	2.40	4.03	5.96
+ distilled router	0.12	0.26	0.58	1.43	2.58	4.01
Peak prefill memory (GB) ↓						
Transformer	4.75	6.80	10.89	19.08	27.27	OOM
Mamba2	3.85	4.67	6.30	9.55	12.81	16.07
Gated DeltaNet	3.54	4.08	5.16	7.32	9.49	11.65
Mamba2 Serial Hybrid	4.13	5.25	7.48	11.95	16.42	20.88
Gated DeltaNet Serial Hybrid	3.79	4.60	6.22	9.45	12.69	15.92
Mamba2 Fused Hybrid	4.36	5.61	8.11	13.12	18.12	23.12
AMOR-Mamba2 ♥	13.06	13.43	14.17	15.65	17.12	18.60
+ distilled router	4.06	4.88	6.51	9.77	13.03	16.29
AMOR-Gated DeltaNet ♥	13.06	13.46	14.27	15.88	17.49	19.10
+ distilled router	3.82	4.42	5.63	8.05	10.46	12.88
Decode latency (ms/token) ↓						
Transformer	17.36	29.37	53.28	100.75	148.61	OOM
Mamba2	16.24	16.26	16.13	16.14	16.12	16.15
Gated DeltaNet	22.57	22.76	22.63	22.56	22.59	22.45
Mamba2 Serial Hybrid	16.58	18.12	21.21	27.30	33.35	39.42
Gated DeltaNet Serial Hybrid	21.13	21.03	21.01	20.95	23.99	29.98
Mamba2 Fused Hybrid	18.19	19.78	22.80	28.92	34.97	41.07
AMOR-Mamba2 ♥	19.22	19.48	22.67	26.81	32.00	34.21
+ distilled router	18.82	18.90	22.11	26.32	31.27	33.73
AMOR-Gated DeltaNet ♥	25.94	26.10	27.53	31.90	37.57	42.08
+ distilled router	25.07	25.34	27.08	31.26	36.46	41.57

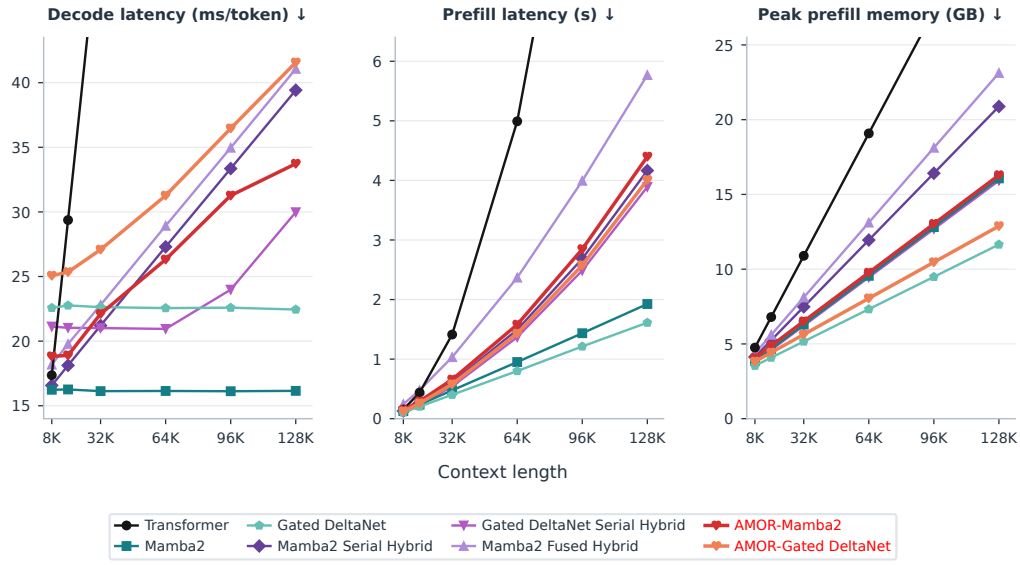

Figure 11: **Long-context inference efficiency.** 1.5B; all eight models on one H100 PCIe. AMOR uses the distilled router. Transformer curves exceed the axis ranges and reach OOM at 128K.

Tables 11–13 report inference efficiency for AMOR with the native entropy gate and the distilled router alongside all baselines at three scales. The distilled router lowers prefill latency and peak prefill memory across scales, while decode gains vary by backbone and context length. In decode latency, AMOR-Mamba2 with the distilled router overtakes the fused hybrid between 8K and 16K at every scale. Its crossover with the serial hybrid occurs at 64K–96K (180M), 16K–32K (440M) and 32K–64K (1.5B).

Table 12: **Inference efficiency.** 180M; all eight architectures on one H100 PCIe, using the protocol of Table 11.

Model	Context length					
	8K	16K	32K	64K	96K	128K
Prefill latency (s) ↓						
Transformer	0.03	0.08	0.25	0.90	1.94	3.48
Mamba2	0.03	0.04	0.07	0.14	0.21	0.28
Gated DeltaNet	0.03	0.04	0.07	0.13	0.19	0.25
Mamba2 Serial Hybrid	0.03	0.05	0.11	0.27	0.51	0.81
Gated DeltaNet Serial Hybrid	0.02	0.04	0.10	0.26	0.49	0.80
Mamba2 Fused Hybrid	0.03	0.06	0.14	0.37	0.70	1.16
AMOR-Mamba2 ♥	0.14	0.27	0.55	1.20	1.96	2.84
+ distilled router	0.03	0.06	0.14	0.38	0.72	1.17
AMOR-Gated DeltaNet ♥	0.14	0.26	0.54	1.19	1.94	2.80
+ distilled router	0.03	0.06	0.13	0.36	0.70	1.14
Peak prefill memory (GB) ↓						
Transformer	0.84	1.29	2.19	3.99	5.79	7.59
Mamba2	0.71	1.02	1.64	2.89	4.13	5.37
Gated DeltaNet	0.59	0.79	1.20	2.01	2.82	3.63
Mamba2 Serial Hybrid	0.83	1.23	2.03	3.62	5.22	6.81
Gated DeltaNet Serial Hybrid	0.70	0.98	1.53	2.65	3.76	4.87
Mamba2 Fused Hybrid	0.92	1.39	2.34	4.24	6.14	8.03
AMOR-Mamba2 ♥	10.11	10.25	10.53	11.08	11.64	12.19
+ distilled router	0.83	1.14	1.77	3.01	4.25	5.50
AMOR-Gated DeltaNet ♥	10.12	10.27	10.57	11.18	11.78	12.39
+ distilled router	0.74	0.97	1.42	2.33	3.23	4.14
Decode latency (ms/token) ↓						
Transformer	7.13	12.40	22.50	42.83	62.50	80.44
Mamba2	8.08	8.17	8.09	8.11	8.14	8.10
Gated DeltaNet	11.63	11.59	11.55	11.62	11.57	11.59
Mamba2 Serial Hybrid	8.05	8.97	10.75	14.25	17.70	21.19
Gated DeltaNet Serial Hybrid	10.53	10.52	10.51	10.53	12.54	15.97
Mamba2 Fused Hybrid	9.68	11.10	13.70	19.00	24.20	29.02
AMOR-Mamba2 ♥	10.50	10.55	12.65	14.43	20.38	21.85
+ distilled router	10.19	10.14	11.89	15.22	16.80	18.16
AMOR-Gated DeltaNet ♥	13.97	13.94	15.15	16.52	18.82	19.98
+ distilled router	13.73	13.88	15.88	16.41	17.87	19.99

Table 13: **Inference efficiency.** 440M; all eight architectures on one H100 PCIe, using the protocol of Table 11.

Model	Context length					
	8K	16K	32K	64K	96K	128K
Prefill latency (s) ↓						
Transformer	0.07	0.21	0.70	2.52	5.49	9.56
Mamba2	0.07	0.11	0.20	0.45	0.60	0.80
Gated DeltaNet	0.05	0.09	0.17	0.34	0.51	0.68
Mamba2 Serial Hybrid	0.06	0.12	0.26	0.66	1.21	1.88
Gated DeltaNet Serial Hybrid	0.05	0.11	0.24	0.61	1.13	1.78
Mamba2 Fused Hybrid	0.10	0.18	0.38	0.90	1.55	2.35
AMOR-Mamba2 ♥	0.18	0.35	0.71	1.56	2.56	3.68
+ distilled router	0.07	0.13	0.29	0.71	1.29	1.99
AMOR-Gated DeltaNet ♥	0.16	0.33	0.68	1.51	2.48	3.58
+ distilled router	0.06	0.12	0.26	0.65	1.20	1.88
Peak prefill memory (GB) ↓						
Transformer	1.85	2.87	4.91	9.00	13.08	17.16
Mamba2	1.36	1.77	2.59	4.24	5.89	7.53
Gated DeltaNet	1.19	1.46	2.00	3.08	4.16	5.24
Mamba2 Serial Hybrid	1.52	2.09	3.21	5.46	7.71	9.96
Gated DeltaNet Serial Hybrid	1.34	1.74	2.55	4.17	5.79	7.41
Mamba2 Fused Hybrid	1.60	2.23	3.49	6.01	8.53	11.05
AMOR-Mamba2 ♥	10.71	10.89	11.26	12.00	12.74	13.48
+ distilled router	1.49	1.90	2.72	4.37	6.02	7.66
AMOR-Gated DeltaNet ♥	10.71	10.91	11.31	12.12	12.93	13.73
+ distilled router	1.36	1.66	2.26	3.47	4.68	5.89
Decode latency (ms/token) ↓						
Transformer	14.87	25.63	46.99	89.82	132.80	175.66
Mamba2	16.07	16.08	16.16	16.17	16.17	16.16
Gated DeltaNet	22.53	22.61	22.51	22.51	22.46	22.44
Mamba2 Serial Hybrid	15.88	17.35	20.16	25.38	31.03	36.49
Gated DeltaNet Serial Hybrid	21.34	21.47	21.46	21.51	21.58	25.80
Mamba2 Fused Hybrid	17.42	18.87	21.64	27.11	32.50	37.99
AMOR-Mamba2 ♥	18.33	18.03	19.12	25.98	22.85	27.10
+ distilled router	17.98	17.69	18.61	25.15	21.37	26.62
AMOR-Gated DeltaNet ♥	24.65	24.87	27.33	28.43	30.62	37.13
+ distilled router	24.68	24.96	26.57	28.76	30.46	36.97

F.4 MEMORY USAGE

At 1.5B, three-attention-layer hybrids use one eighth of the Transformer's KV storage: 3.22 versus 25.77 GB at 128K (Table 14). Every position contributes keys and values regardless of firing. Recurrent-state storage is fixed with context length: 26.00 MB for 24 Mamba2 layers and 20.05 MB for 24 Gated DeltaNet layers; serial hybrids use 21 recurrent layers.

Table 14: **KV cache and recurrent-state storage.** 1.5B; bfloat16 caches and three attention layers per hybrid. N/A denotes an unused cache or state.

Model	**Recurrent state size**	**Key-value cache** (GB) ↓				
	MB, fixed	3K	8K	16K	32K	128K
Transformer	N/A	0.60	1.61	3.22	6.44	25.77
Mamba2	26.00	N/A	N/A	N/A	N/A	N/A
Gated DeltaNet	20.05	N/A	N/A	N/A	N/A	N/A
Mamba2 Serial Hybrid	22.75	0.08	0.20	0.40	0.81	3.22
Gated DeltaNet Serial Hybrid	17.54	0.08	0.20	0.40	0.81	3.22
Mamba2 Fused Hybrid	26.00	0.08	0.20	0.40	0.81	3.22
AMOR-Mamba2 ♥	26.00	0.08	0.20	0.40	0.81	3.22
AMOR-Gated DeltaNet ♥	20.05	0.08	0.20	0.40	0.81	3.22

F.5 DECODE COST AND BREAK-EVEN

All costs below are matrix-multiply FLOPs per decoded token and per block, counting scalar multiplication and addition as one floating-point operation each. Let D be model width, V vocabulary size, L the number of attended cache positions (including the current token), and $0 \le f < 1$ a fixed firing rate. Here $\mathbf{A}$ denotes the softmax attention weights.

Native entropy gate decode vs. always-on attention. For otherwise identical attention blocks, AMOR skips the following when $g_t = 0$:

- **Attention scores $\mathbf{QK}^\top$:** $2DL$ FLOPs, summed across heads; the subsequent softmax is also skipped.
- **Weighted values $\mathbf{AV}$:** $2DL$ FLOPs, summed across heads.
- **Query projection:** $2D^2$ FLOPs; RoPE on $\mathbf{Q}$ is also skipped.
- **Output projection $\mathbf{W}_O$:** $2D^2$ FLOPs.

Key/value projections and cache updates always run. The native entropy gate adds $C_{\text{lm_head}} = 2VD$ matrix-multiply FLOPs.

Attention matmul savings only. Counting the skipped $\mathbf{QK}^\top$ and $\mathbf{AV}$ operations gives:

$$\begin{aligned} C_{\text{attn}}(L) &= 4DL, \\ (1-f)\, C_{\text{attn}}(L) &> C_{\text{lm_head}}, \\ L &> \frac{V}{2(1-f)}. \end{aligned} \tag{10}$$

Including query and output projection savings. Adding their $4D^2$ FLOPs to the skipped cost gives:

$$\begin{aligned} (1-f)(4DL + 4D^2) &> C_{\text{lm_head}}, \\ L &> \frac{V}{2(1-f)} - D. \end{aligned} \tag{11}$$

At $V = 128{,}256$, $D = 2048$ and an illustrative $f = 0.4$, the equality thresholds are 106,880 tokens in Eq. 10 and 104,832 in Eq. 11. These are arithmetic thresholds, not measured latency crossovers.

Distilled router vs. native entropy gate. The width-r predictor replaces the vocabulary projection with three linear maps:

- Hidden projection, $D \to r$: $2Dr$ FLOPs.
- Scalar output, $r \to 1$: $2r$ FLOPs.
- Linear skip, $D \to 1$: $2D$ FLOPs.

For the evaluated 1.5B models, $D = 2048$, $V = 128{,}256$ and $r = 512$:

$$\begin{aligned} C_{\text{route}} &= 2Dr + 2r + 2D = 2{,}102{,}272, \\ C_{\text{lm_head}} &= 2VD = 525{,}336{,}576, \\ \frac{C_{\text{lm_head}}}{C_{\text{route}}} &\approx 250. \end{aligned} \tag{12}$$

This ratio measures gating matmul savings, not total model speedup. Both comparisons omit softmax, entropy reduction and other non-matmul work; measured latency also depends on memory traffic, kernels, synchronization and architecture (Appendix F.3; Fig. 11).

G Distilled Entropy Router

G.1 Definition and fitting

The deployment ablation replaces each block's LM-head evaluation with a predictor fitted on the frozen pretrained model:

$$\widehat{\mathcal{H}}_\ell(\tilde{h}_t) = a_\ell^\top \mathrm{SiLU}(W_\ell \tilde{h}_t + b_\ell) + s_\ell^\top \tilde{h}_t + c_\ell, \qquad W_\ell \in \mathbb{R}^{512 \times D}. \tag{13}$$

Here W_ℓ and b_ℓ parameterize the hidden projection, a_ℓ and c_ℓ the scalar output, and s_ℓ the linear skip. The gate remains $g_t = \mathbf{1}[\widehat{\mathcal{H}}_\ell(\tilde{h}_t) > \tau_\ell]$ with the frozen entropy threshold. Each predictor has a 512-unit hidden layer at all scales. The three predictors add 1.19M / 1.58M / 3.15M parameters at 180M / 440M / 1.5B.

At each scale, we fit the predictor for Block 0 first, then fit Blocks 1 and 2 with earlier predictors enabled. FineWeb-Edu, PG-19 and CodeParrot provide disjoint fitting, calibration and evaluation regions with about 1.57M, 0.295M and 0.590M positions per block, respectively. We initialize the linear skip with ridge regression, fit with boundary-weighted SmoothL1 loss for 30 epochs (learning rate 10^{-3}, batch size 16,384), and calibrate the output bias on the held-out calibration split. Predictors run in float32 within the model's bfloat16 execution.

G.2 Downstream quality

Table 15 summarizes paired results with the native entropy gate and distilled router on the same frozen pretrained checkpoints. Across scales, common-sense averages change by at most 0.12 points and LongBench averages by less than 0.08 points. At 1.5B, the largest change across all four evaluation suites is 0.13 points. Forward FLOPs are calculated for 3K inputs, replacing the three LM-head evaluation costs with predictor costs for the distilled router (Appendix F.5).

Table 15: **Native entropy gate vs. distilled router: downstream performance and forward FLOPs.** 180M, 440M and 1.5B. Evaluation settings follow Appendix E. Δ = distilled router − native entropy gate.

Model	Gate	Common-sense avg acc (%) ↑	Cloze retrieval avg (%) ↑	S-NIAH avg acc (%) ↑	LongBench avg score (%) ↑	Fwd FLOPs GFLOPs/tok ↓
180M						
AMOR-Mamba2 ♥	Native entropy gate	35.82	N/A	N/A	5.89	0.96
	Distilled router	35.84	N/A	N/A	5.91	0.38
	Δ	+0.02	N/A	N/A	+0.02	-0.59
AMOR-Gated DeltaNet ♥	Native entropy gate	35.67	N/A	N/A	6.12	0.98
	Distilled router	35.62	N/A	N/A	6.05	0.39
	Δ	-0.05	N/A	N/A	-0.07	-0.59
440M						
AMOR-Mamba2 ♥	Native entropy gate	39.58	N/A	N/A	6.76	1.63
	Distilled router	39.55	N/A	N/A	6.83	0.85
	Δ	-0.03	N/A	N/A	+0.06	-0.78
AMOR-Gated DeltaNet ♥	Native entropy gate	40.45	N/A	N/A	6.95	1.71
	Distilled router	40.56	N/A	N/A	7.00	0.93
	Δ	+0.10	N/A	N/A	+0.05	-0.78
1.5B						
AMOR-Mamba2 ♥	Native entropy gate	45.81	27.13	74.11	8.27	4.35
	Distilled router	45.76	27.04	74.16	8.25	2.78
	Δ	-0.06	-0.09	+0.04	-0.02	-1.57
AMOR-Gated DeltaNet ♥	Native entropy gate	45.13	30.01	77.78	8.44	4.70
	Distilled router	45.01	30.14	77.67	8.43	3.13
	Δ	-0.12	+0.13	-0.11	-0.01	-1.57

Tables 16–18 give per-task results in the format of Tables 1–3. Retrieval and S-NIAH follow the 1.5B evaluation setting in Appendix E.2.

Table 16: **Common-sense routing comparison.** 180M, 440M and 1.5B.

Model	FW-Edu PPL ↓	LAMB PPL ↓	LAMB acc ↑	HSwag acc ↑	PIQA acc ↑	ARC-E acc ↑	ARC-C acc ↑	WinoG acc ↑	OBQA acc ↑	TQA mc2 acc ↑	Avg acc ↑
180M											
AMOR-Mamba2 ♥	31.46	240.27	20.3	27.4	60.5	46.5	18.0	51.3	16.6	45.9	35.8
+ distilled router	N/A	250.77	20.3	27.3	60.7	46.6	17.7	51.6	16.8	45.6	35.8
AMOR-Gated DeltaNet ♥	30.60	234.20	19.5	27.3	61.0	47.7	17.3	51.1	16.2	45.2	35.7
+ distilled router	N/A	237.83	19.4	27.3	60.9	47.7	17.5	51.0	15.8	45.3	35.6
440M											
AMOR-Mamba2 ♥	19.60	57.37	29.3	31.4	65.9	55.4	24.7	52.4	21.0	36.4	39.6
+ distilled router	N/A	57.51	29.3	31.5	65.8	55.6	24.8	52.2	21.2	36.0	39.6
AMOR-Gated DeltaNet ♥	19.27	50.15	30.2	31.8	65.0	56.0	24.6	51.3	23.8	40.9	40.5
+ distilled router	N/A	50.30	30.2	31.8	65.2	56.0	25.2	51.6	23.8	40.7	40.6
1.5B											
AMOR-Mamba2 ♥	13.30	23.01	39.2	38.6	70.9	67.0	30.2	54.7	26.8	39.0	45.8
+ distilled router	N/A	23.02	39.2	38.6	70.5	66.8	30.1	55.0	26.8	39.0	45.8
AMOR-Gated DeltaNet ♥	13.10	21.49	39.1	39.2	70.6	66.4	30.5	53.7	24.8	36.9	45.1
+ distilled router	N/A	21.54	39.0	39.2	70.2	66.3	30.5	53.2	24.8	36.8	45.0

Table 17: **Retrieval and S-NIAH routing comparison.** 1.5B.

Model	SWDE↑	SQuAD↑	FDA↑	TQA↑	NQ↑	DROP↑	S-NIAH-1↑			S-NIAH-2↑			S-NIAH-3↑		
Context length	2048						1024	2048	4096	1024	2048	4096	1024	2048	4096
AMOR-Mamba2 ♥	33.8	36.0	21.2	40.4	10.2	21.1	100.0	95.8	44.4	100.0	99.6	47.2	70.8	83.6	25.6
+ distilled router	33.7	36.2	21.6	39.8	10.1	20.9	100.0	95.8	44.8	100.0	99.6	47.2	71.2	83.4	25.4
AMOR-Gated DeltaNet ♥	47.4	36.8	24.9	43.1	10.5	17.4	100.0	96.2	48.0	100.0	98.6	53.0	88.8	86.0	29.4
+ distilled router	47.6	37.0	25.4	42.5	10.7	17.7	100.0	96.2	48.0	100.0	98.4	53.0	89.0	86.0	28.4

Table 18: **LongBench routing comparison.** 180M, 440M and 1.5B.

Model	Single-Doc QA↑			Multi-Doc QA↑			Summarization↑			Few-shot↑			Code↑		Avg↑
	NQA	QQA	MFQ	HQA	2WM	Mus	GvR	QMS	MNs	TRC	TQA	SSM	LCC	RBP	
180M															
AMOR-Mamba2 ♥	1.0±0.15	2.0±0.24	5.9±0.58	1.8±0.36	2.5±0.46	1.7±0.31	5.7±0.25	11.5±0.41	8.5±0.30	1.5±0.86	6.8±0.85	8.3±0.44	14.0±0.45	11.2±0.32	5.9±0.13
+ distilled router	1.1±0.17	2.2±0.26	5.7±0.57	1.8±0.36	2.6±0.47	1.7±0.31	5.7±0.25	12.1±0.42	8.3±0.29	1.5±0.86	6.7±0.84	8.0±0.46	14.0±0.45	11.3±0.34	5.9±0.13
AMOR-Gated DeltaNet ♥	0.8±0.11	2.0±0.25	4.6±0.51	1.6±0.31	1.8±0.36	1.6±0.31	4.4±0.23	11.5±0.44	7.4±0.31	8.8±1.99	6.8±0.63	8.3±0.51	12.0±0.31	14.0±0.32	6.1±0.17
+ distilled router	0.8±0.11	2.1±0.27	4.5±0.51	1.5±0.32	2.1±0.41	1.6±0.29	4.4±0.23	11.4±0.46	7.2±0.31	8.2±1.93	6.6±0.62	8.4±0.51	11.8±0.31	14.0±0.32	6.1±0.17
440M															
AMOR-Mamba2 ♥	1.3±0.14	2.3±0.22	7.5±0.77	2.6±0.44	4.5±0.59	2.1±0.39	7.3±0.24	12.6±0.47	11.5±0.36	5.0±1.54	9.0±1.15	9.2±0.54	9.8±0.33	10.1±0.32	6.8±0.17
+ distilled router	1.2±0.14	2.4±0.23	7.5±0.76	2.9±0.49	4.6±0.59	2.1±0.39	7.5±0.25	13.0±0.46	11.6±0.37	4.5±1.47	8.9±1.15	9.1±0.55	10.0±0.33	10.3±0.33	6.8±0.17
AMOR-Gated DeltaNet ♥	2.0±0.33	3.2±0.31	10.6±0.90	3.3±0.48	6.6±0.72	2.0±0.32	6.9±0.22	13.0±0.51	7.5±0.46	6.0±1.68	8.3±0.84	8.9±0.56	9.2±0.26	9.7±0.24	7.0±0.18
+ distilled router	2.0±0.33	3.2±0.29	10.6±0.90	3.3±0.48	6.7±0.73	2.1±0.35	7.1±0.23	13.1±0.49	7.6±0.47	6.5±1.75	8.2±0.80	8.7±0.55	9.3±0.26	9.8±0.24	7.0±0.18
1.5B															
AMOR-Mamba2 ♥	1.6±0.15	3.3±0.32	10.1±0.83	2.9±0.50	6.1±0.71	1.8±0.37	7.6±0.23	14.7±0.46	13.2±0.47	13.0±2.38	13.2±1.49	7.5±0.51	9.3±0.26	11.5±0.29	8.3±0.23
+ distilled router	1.6±0.15	3.2±0.31	10.1±0.83	3.0±0.53	5.8±0.70	1.8±0.34	7.6±0.23	14.4±0.46	13.3±0.46	13.0±2.38	13.3±1.45	7.4±0.49	9.5±0.27	11.5±0.29	8.3±0.23
AMOR-Gated DeltaNet ♥	1.7±0.17	4.0±0.35	11.4±0.87	3.6±0.49	7.0±0.74	1.9±0.32	8.1±0.30	15.1±0.50	10.8±0.43	11.0±2.22	11.8±1.22	11.4±0.68	10.1±0.38	10.3±0.41	8.4±0.22
+ distilled router	1.7±0.17	4.0±0.36	11.1±0.88	3.8±0.51	6.8±0.74	1.9±0.32	8.1±0.30	15.4±0.50	10.9±0.41	11.0±2.22	11.4±1.22	11.4±0.67	10.1±0.37	10.4±0.41	8.4±0.22

Tables 19–24 report length extrapolation results across all three scales (Appendix E.3.2). FineWeb-Edu uses the same protocol with additional 1K and 2K contexts. At 20K, relative differences remain within 1.62% on six of seven domains. CodeParrot is the exception: the distilled router lowers perplexity for both AMOR variants at every scale, including 979.92 to 475.75 for AMOR-Mamba2 and 289.14 to 203.17 for AMOR-Gated DeltaNet at 440M. Larger differences also occur at shorter contexts: 180M AMOR-Mamba2's PG-19 perplexity at 3K changes from 106.05 to 87.37.

Table 19: **Length extrapolation routing comparison.** 180M.

Model	Context length; PPL ↓									
	3K	4K	6K	8K	10K	12K	14K	16K	18K	20K
GovReport										
AMOR-Mamba2 ♥	24.72	24.27	24.95	26.13	26.72	27.63	28.43	28.85	29.05	29.34
+ distilled router	25.14	24.58	25.20	26.38	26.93	27.87	28.72	29.17	29.40	29.68
AMOR-Gated DeltaNet ♥	23.14	22.53	22.93	23.57	23.56	23.97	24.33	24.31	24.29	24.28
+ distilled router	23.42	22.77	23.14	23.76	23.73	24.14	24.48	24.46	24.43	24.42
QMSum										
AMOR-Mamba2 ♥	29.20	31.71	36.29	38.51	41.38	44.67	46.26	45.61	46.51	50.10
+ distilled router	29.39	32.05	36.59	38.62	41.06	44.08	45.57	44.92	45.93	49.63
AMOR-Gated DeltaNet ♥	28.22	30.14	33.64	35.39	37.92	40.22	40.50	39.14	39.12	41.63
+ distilled router	28.34	30.30	33.77	35.41	37.80	40.02	40.27	38.93	38.93	41.46
NarrativeQA										
AMOR-Mamba2 ♥	159.22	160.34	164.00	165.90	152.32	143.05	137.83	133.96	131.37	129.06
+ distilled router	154.98	153.30	155.65	156.90	144.39	137.06	133.64	131.36	129.90	128.43
AMOR-Gated DeltaNet ♥	146.35	145.54	147.09	146.89	132.77	122.98	117.35	112.55	109.26	106.30
+ distilled router	145.61	143.67	144.28	143.42	129.08	119.81	114.71	110.46	107.68	105.05
Qasper										
AMOR-Mamba2 ♥	54.66	54.89	54.16	56.99	58.99	60.79	62.88	64.50	66.34	68.26
+ distilled router	50.30	50.78	50.73	54.10	56.58	58.64	61.22	63.18	65.52	67.64
AMOR-Gated DeltaNet ♥	51.17	51.33	49.31	50.71	51.37	51.65	52.57	52.71	53.45	54.12
+ distilled router	49.19	49.58	47.83	49.40	50.24	50.66	51.70	51.94	52.80	53.55
CodeParrot										
AMOR-Mamba2 ♥	314.18	369.89	427.68	688.59	1045.73	1547.90	2152.41	2530.52	2855.07	3080.56
+ distilled router	303.74	324.28	311.97	372.18	426.20	503.95	610.43	674.70	760.62	845.78
AMOR-Gated DeltaNet ♥	393.21	431.41	416.22	508.64	595.98	728.60	876.08	945.24	1017.08	1068.88
+ distilled router	383.10	396.98	348.64	376.57	395.67	445.66	492.60	505.26	532.69	559.38
PG-19										
AMOR-Mamba2 ♥	106.05	89.15	83.93	93.12	96.37	96.45	96.04	94.77	95.69	94.77
+ distilled router	87.37	79.33	79.48	90.26	94.36	95.05	95.18	94.24	95.44	94.85
AMOR-Gated DeltaNet ♥	91.00	80.41	77.95	85.40	87.65	86.94	86.31	84.32	84.14	82.57
+ distilled router	82.15	74.79	74.83	83.14	85.91	85.65	85.35	83.57	83.50	82.00

Table 20: **FineWeb-Edu length extrapolation routing comparison.** 180M.

Model	Context length; PPL ↓											
	1K	2K	3K	4K	6K	8K	10K	12K	14K	16K	18K	20K
FineWeb-Edu												
AMOR-Mamba2 ♥	32.91	28.07	28.83	29.35	29.31	29.54	32.67	32.18	32.35	32.18	32.17	31.90
+ distilled router	33.02	28.24	29.00	29.55	29.51	29.74	32.89	32.38	32.55	32.39	32.42	32.15
AMOR-Gated DeltaNet ♥	32.21	27.42	28.12	28.55	28.34	28.41	31.24	30.56	30.59	30.25	30.14	29.83
+ distilled router	32.33	27.54	28.24	28.68	28.48	28.55	31.36	30.67	30.71	30.35	30.25	29.94

Table 21: **Length extrapolation routing comparison.** 440M.

Model	Context length; PPL ↓									
	3K	4K	6K	8K	10K	12K	14K	16K	18K	20K
GovReport										
AMOR-Mamba2 ♥	15.10	14.78	15.11	15.80	16.17	16.80	17.36	17.69	17.83	18.06
+ distilled router	15.17	14.81	15.13	15.83	16.20	16.85	17.43	17.77	17.92	18.17
AMOR-Gated DeltaNet ♥	14.70	14.35	14.51	14.72	14.64	14.80	14.96	14.92	14.91	14.94
+ distilled router	14.74	14.40	14.53	14.73	14.64	14.80	14.97	14.91	14.90	14.94
QMSum										
AMOR-Mamba2 ♥	19.71	21.13	24.03	26.25	29.17	32.06	33.70	33.41	34.71	38.26
+ distilled router	19.70	21.19	24.07	26.22	29.10	31.98	33.64	33.40	34.72	38.33
AMOR-Gated DeltaNet ♥	19.33	20.64	22.85	23.78	25.05	26.57	26.68	25.67	25.58	26.92
+ distilled router	19.36	20.68	22.86	23.76	25.02	26.52	26.64	25.65	25.58	26.95
NarrativeQA										
AMOR-Mamba2 ♥	92.51	92.96	95.95	97.95	89.49	83.61	80.57	78.38	77.14	76.18
+ distilled router	90.35	89.51	91.90	93.75	86.02	81.28	79.10	77.51	76.77	76.15
AMOR-Gated DeltaNet ♥	86.58	85.33	84.97	83.62	74.31	68.22	64.54	61.97	60.25	58.79
+ distilled router	86.04	84.28	83.35	81.58	72.47	66.84	63.55	61.30	59.80	58.49
Qasper										
AMOR-Mamba2 ♥	32.21	32.31	31.79	33.52	34.92	36.30	37.71	39.05	40.51	41.63
+ distilled router	30.26	30.50	30.26	32.18	33.74	35.31	36.89	38.35	39.98	41.17
AMOR-Gated DeltaNet ♥	30.93	30.90	29.58	30.04	30.28	30.25	30.67	30.73	31.23	31.65
+ distilled router	29.83	29.96	28.84	29.36	29.69	29.72	30.20	30.31	30.88	31.35
CodeParrot										
AMOR-Mamba2 ♥	142.47	171.03	188.98	275.66	378.98	531.07	702.68	796.29	899.08	979.92
+ distilled router	137.46	147.19	142.72	173.52	209.09	265.56	332.95	371.48	422.99	475.75
AMOR-Gated DeltaNet ♥	158.09	173.46	166.35	188.63	199.46	224.47	251.23	260.66	276.53	289.14
+ distilled router	154.91	158.52	140.95	148.50	151.28	164.88	178.38	181.91	192.72	203.17
PG-19										
AMOR-Mamba2 ♥	57.90	49.16	48.45	53.19	54.84	54.89	54.61	54.04	54.72	53.87
+ distilled router	49.65	45.28	46.41	51.90	54.06	54.50	54.48	54.02	54.77	54.07
AMOR-Gated DeltaNet ♥	50.80	45.05	45.11	48.63	49.47	48.89	48.23	47.22	47.23	46.24
+ distilled router	47.57	43.27	44.09	47.89	48.91	48.51	47.97	47.01	47.09	46.17

Table 22: **FineWeb-Edu length extrapolation routing comparison.** 440M.

Model	Context length; PPL ↓											
	1K	2K	3K	4K	6K	8K	10K	12K	14K	16K	18K	20K
FineWeb-Edu												
AMOR-Mamba2 ♥	20.43	17.80	18.28	18.65	18.45	18.57	20.46	20.17	20.29	20.22	20.30	20.13
+ distilled router	20.45	17.83	18.32	18.68	18.48	18.60	20.49	20.20	20.31	20.24	20.31	20.14
AMOR-Gated DeltaNet ♥	20.18	17.54	17.99	18.32	18.02	17.98	19.59	19.13	19.14	18.96	18.90	18.72
+ distilled router	20.19	17.56	18.00	18.33	18.03	17.98	19.60	19.14	19.14	18.96	18.91	18.73

Table 23: **Length extrapolation routing comparison.** 1.5B.

Model	Context length; PPL ↓									
	3K	4K	6K	8K	10K	12K	14K	16K	18K	20K
GovReport										
AMOR-Mamba2 ♥	11.13	10.93	11.20	11.73	12.06	12.60	13.15	13.52	13.67	13.92
+ distilled router	11.14	10.91	11.17	11.69	12.04	12.60	13.17	13.55	13.73	13.99
AMOR-Gated DeltaNet ♥	10.88	10.63	10.76	11.02	11.08	11.36	11.58	11.67	11.73	11.87
+ distilled router	10.91	10.65	10.78	11.03	11.08	11.36	11.58	11.66	11.73	11.86
QMSum										
AMOR-Mamba2 ♥	14.73	15.82	18.18	19.87	22.18	25.54	27.84	27.84	29.80	34.10
+ distilled router	14.71	15.84	18.17	19.84	22.20	25.68	28.08	28.15	30.19	34.65
AMOR-Gated DeltaNet ♥	14.60	15.78	18.16	19.82	22.12	24.62	25.73	24.88	25.56	28.16
+ distilled router	14.61	15.81	18.19	19.83	22.16	24.71	25.87	25.05	25.73	28.40
NarrativeQA										
AMOR-Mamba2 ♥	62.36	61.73	63.07	64.81	60.75	58.34	57.36	56.95	56.68	56.76
+ distilled router	60.91	59.78	61.06	62.78	59.36	57.61	57.09	57.05	57.01	57.24
AMOR-Gated DeltaNet ♥	59.47	58.72	59.57	60.24	55.44	52.22	50.28	48.99	48.10	47.35
+ distilled router	58.97	57.84	58.36	58.76	54.10	51.17	49.61	48.60	47.93	47.32
Qasper										
AMOR-Mamba2 ♥	22.01	22.12	21.82	22.86	24.10	25.25	26.51	27.68	28.94	29.83
+ distilled router	21.15	21.28	21.14	22.31	23.67	24.93	26.33	27.60	28.99	29.97
AMOR-Gated DeltaNet ♥	21.15	21.21	20.82	21.68	22.66	23.42	24.17	24.91	25.65	26.26
+ distilled router	20.53	20.57	20.29	21.20	22.26	23.11	23.95	24.74	25.59	26.29
CodeParrot										
AMOR-Mamba2 ♥	89.17	101.90	113.85	161.26	218.54	313.57	417.63	482.96	561.35	627.21
+ distilled router	85.64	88.05	87.51	111.80	144.48	201.19	266.65	306.95	363.99	421.59
AMOR-Gated DeltaNet ♥	81.74	89.97	95.00	122.66	155.49	210.90	267.07	304.26	343.55	376.96
+ distilled router	79.02	81.59	80.79	99.06	121.05	156.79	193.35	215.94	242.84	269.15
PG-19										
AMOR-Mamba2 ♥	38.16	33.17	33.24	36.16	37.41	37.59	37.51	37.35	38.22	37.73
+ distilled router	34.14	31.13	32.24	35.64	37.16	37.54	37.60	37.52	38.45	38.07
AMOR-Gated DeltaNet ♥	37.14	31.90	31.94	34.50	35.26	35.01	34.62	34.18	34.63	34.00
+ distilled router	33.30	30.26	31.18	34.07	35.02	34.90	34.59	34.20	34.69	34.12

Table 24: **FineWeb-Edu length extrapolation routing comparison.** 1.5B.

Model	Context length; PPL ↓											
	1K	2K	3K	4K	6K	8K	10K	12K	14K	16K	18K	20K
FineWeb-Edu												
AMOR-Mamba2 ♥	14.64	12.88	13.08	13.30	13.23	13.30	14.11	14.03	14.26	14.33	14.50	14.42
+ distilled router	14.65	12.89	13.09	13.30	13.23	13.31	14.12	14.03	14.26	14.33	14.50	14.41
AMOR-Gated DeltaNet ♥	14.46	12.73	12.91	13.11	12.98	13.00	13.71	13.56	13.69	13.68	13.76	13.64
+ distilled router	14.47	12.74	12.91	13.11	12.98	13.00	13.71	13.56	13.69	13.67	13.76	13.63

G.3 Routing-signal controls

Table 25 compares the native entropy gate and distilled router with a random router and zeroed attention outputs at 1.5B. The random router is an untrained predictor with a 256-unit hidden layer and an uncalibrated firing rate. Attention output zeroing sets the output projection $\mathbf{W}_O$ to zero in each of the $K = 3$ AMOR blocks, keeping all other weights fixed. Random routing lowers common-sense accuracy (Appendix E.1) by about three points and three-task retrieval (SWDE, SQuAD-completion and FDA) by roughly two thirds. Most of the common-sense loss is on LAMBADA: AMOR-Mamba2 accuracy falls from 39.2% to 13.2%, and perplexity rises from 23.01 to 691.84. Attention output zeroing lowers the retrieval averages to 9.0% and 11.4% on AMOR-Mamba2 and AMOR-Gated DeltaNet.

Table 25: **Routing-signal controls.** 1.5B.

Routing mechanism	AMOR-Mamba2 ♥		AMOR-Gated DeltaNet ♥	
	Common-sense avg acc (%) ↑	**Cloze retrieval** avg (%) ↑	**Common-sense** avg acc (%) ↑	**Cloze retrieval** avg (%) ↑
AMOR (native entropy gate)	45.8	30.4	45.1	36.4
+ distilled router	45.8	30.5	45.0	36.7
Random router	42.4	9.8	42.3	11.8
Attention output zeroed	N/A	9.0	N/A	11.4

H Design Ablations

We study six design ablations of AMOR, each pretrained from scratch with its modified architecture. Within each scale, runs share the model width, FineWeb-Edu data, tokenizer, sequence length, training-token budget, effective batch size, AdamW settings, learning-rate schedule and warmup, gradient clipping, bfloat16 precision and training seed 42, following Section 4 and Appendix C.

Block design. We compare canonical AMOR with three variants on both recurrent backbones. **Symmetric block normalization** (AMOR-symmetric) uses the backbone's final RMSNorm as Block 0's pre-norm and carries an unnormalized residual stream through all three AMOR blocks. The **ungated post-hoc hybrid** (AMOR-ungated) uses this symmetric layout with always-on causal attention, plain residual addition and Gaussian-initialized output projections. **AMOR-MLP** adds an unconditional, pre-norm SwiGLU MLP after each AMOR block, matching the backbone's MLP hidden dimension.

At 440M, canonical AMOR has the highest eight-task common-sense average among these variants on both backbones (Table 26). At 1.5B, symmetric block normalization slightly lowers FW-Edu PPL and has mixed common-sense results (Table 27). It trails canonical AMOR on four of six cloze retrieval tasks with Mamba2 and five with Gated DeltaNet (Table 28).

Block count. We compare $K = 3$ and $\boldsymbol{K = 1}$ **AMOR blocks** after the same recurrent backbone at 180M, 440M and 1.5B. Each block has its own entropy gate, attention projections and per-channel scaling vector $\boldsymbol{\alpha}$; $K = 3$ is the canonical configuration presented in the main paper.

With the Mamba2 backbone, $K = 1$ has a higher common-sense average at 180M and 440M, while $K = 3$ leads at 1.5B. With Gated DeltaNet, $K = 3$ leads at all three scales (Table 29).

Attention-block placement. Table 30 compares canonical AMOR, which appends $K = 3$ blocks after the recurrent backbone, with **AMOR-serial** and **AMOR-fused** at 440M. AMOR-serial replaces recurrent layers at $\{6, 12, 18\}$ with AMOR blocks. AMOR-fused runs AMOR gated attention blocks in parallel with the recurrent mixer at layers $\{0, 12, 23\}$, mirroring the fused hybrid baseline's placement in Section 4. Both evaluate entropy at intermediate depths.

Relative to canonical AMOR, serial placement lowers the common-sense average by 0.90 points with Mamba2 and 0.56 points with Gated DeltaNet. Fused placement leaves the Mamba2 average unchanged and lowers the Gated DeltaNet average by 0.84 points (Table 30).

Table 26: **AMOR block-design ablations: common-sense comparison.** 440M; in the format of Table 1.

Model	LAMB acc ↑	HSwag acc ↑	PIQA acc ↑	ARC-E acc ↑	ARC-C acc ↑	WinoG acc ↑	OBQA acc ↑	TQA mc2 acc ↑	Avg acc ↑
AMOR-Mamba2 ♥	**29.3**	31.4	65.9	55.4	**24.7**	**52.4**	21.0	36.4	**39.6**
AMOR-symmetric-Mamba2	28.3	31.7	65.1	56.5	22.1	49.2	**23.0**	35.7	38.9
AMOR-ungated-Mamba2	28.9	31.5	66.0	55.3	22.9	47.6	21.6	37.1	38.9
AMOR-MLP-Mamba2	27.4	**31.8**	**66.2**	**56.7**	23.1	51.1	19.2	**39.2**	39.3
AMOR-Gated DeltaNet ♥	**30.2**	31.8	65.0	56.0	24.6	**51.3**	**23.8**	40.9	**40.5**
AMOR-symmetric-Gated DeltaNet	27.9	31.7	65.0	**56.9**	**25.1**	50.5	20.0	**42.1**	39.9
AMOR-ungated-Gated DeltaNet	28.6	31.8	65.8	56.4	24.7	49.6	20.0	39.4	39.5
AMOR-MLP-Gated DeltaNet	27.4	**31.9**	**66.0**	55.9	24.6	51.0	20.2	39.3	39.5

Table 27: **Symmetric block normalization ablation: common-sense comparison.** 1.5B; in the format of Table 1.

Model	FW-Edu PPL ↓	LAMB PPL ↓	LAMB acc ↑	HSwag acc ↑	PIQA acc ↑	ARC-E acc ↑	ARC-C acc ↑	WinoG acc ↑	OBQA acc ↑	TQA mc2 acc ↑	Avg acc ↑
1.5B											
AMOR-Mamba2 ♥	13.30	23.01	39.2	38.6	70.9	67.0	30.2	54.7	26.8	39.0	45.8
AMOR-symmetric-Mamba2	13.25	20.79	40.1	38.9	70.6	66.0	31.8	53.0	24.6	38.5	45.4
AMOR-Gated DeltaNet ♥	13.10	21.49	39.1	39.2	70.6	66.4	30.5	53.7	24.8	36.9	45.1
AMOR-symmetric-Gated DeltaNet	13.09	22.64	38.2	39.1	71.4	67.4	32.3	55.4	24.4	35.5	45.5

Table 28: **Symmetric block normalization ablation: retrieval and S-NIAH comparison.** 1.5B; in the format of Table 2.

Model	SWDE↑	SQuAD↑	FDA↑	TQA↑	NQ↑	DROP↑	S-NIAH-1↑			S-NIAH-2↑			S-NIAH-3↑		
Context length	2048						1024	2048	4096	1024	2048	4096	1024	2048	4096
AMOR-Mamba2 ♥	33.8	36.0	21.2	40.4	10.2	21.1	100.0	95.8	44.4	100.0	99.6	47.2	70.8	83.6	25.6
AMOR-symmetric-Mamba2	31.1	35.5	22.1	41.4	9.9	17.2	100.0	92.6	42.2	100.0	90.2	47.6	79.2	85.8	19.2
AMOR-Gated DeltaNet ♥	47.4	36.8	24.9	43.1	10.5	17.4	100.0	96.2	48.0	100.0	98.6	53.0	88.8	86.0	29.4
AMOR-symmetric-Gated DeltaNet	30.4	35.2	9.4	39.2	10.1	18.4	100.0	99.0	75.4	100.0	32.8	29.4	69.2	47.8	19.2

Table 29: **AMOR block-count ablation: common-sense comparison of $K = 3$ and $K = 1$.** 180M, 440M and 1.5B; in the format of Table 1.

Variant	LAMB acc ↑	HSwag acc ↑	PIQA acc ↑	ARC-E acc ↑	ARC-C acc ↑	WinoG acc ↑	OBQA acc ↑	TQA mc2 acc ↑	Avg acc ↑
180M									
AMOR-Mamba2 ♥ (K=3)	20.3	27.4	60.5	46.5	18.0	51.3	16.6	45.9	35.8
AMOR-Mamba2 (K=1)	19.7	27.5	60.7	47.6	18.7	51.9	18.4	44.4	36.1
AMOR-Gated DeltaNet ♥ (K=3)	19.5	27.3	61.0	47.7	17.3	51.1	16.2	45.2	35.7
AMOR-Gated DeltaNet (K=1)	19.0	27.4	60.1	46.7	19.0	51.3	16.6	44.1	35.5
440M									
AMOR-Mamba2 ♥ (K=3)	29.3	31.4	65.9	55.4	24.7	52.4	21.0	36.4	39.6
AMOR-Mamba2 (K=1)	29.2	31.6	65.6	56.2	23.9	50.4	21.6	39.5	39.7
AMOR-Gated DeltaNet ♥ (K=3)	30.2	31.8	65.0	56.0	24.6	51.3	23.8	40.9	40.5
AMOR-Gated DeltaNet (K=1)	28.4	31.9	65.7	56.9	24.3	49.6	21.8	41.7	40.0
1.5B									
AMOR-Mamba2 ♥ (K=3)	39.2	38.6	70.9	67.0	30.2	54.7	26.8	39.0	45.8
AMOR-Mamba2 (K=1)	36.7	38.9	70.2	67.6	30.6	53.4	23.8	34.6	44.5
AMOR-Gated DeltaNet ♥ (K=3)	39.1	39.2	70.6	66.4	30.5	53.7	24.8	36.9	45.1
AMOR-Gated DeltaNet (K=1)	38.4	39.4	71.0	65.7	30.1	53.8	26.6	35.5	45.1

Table 30: **Attention-block placement ablation: common-sense comparison.** 440M; in the format of Table 1.

Model	LAMB acc ↑	HSwag acc ↑	PIQA acc ↑	ARC-E acc ↑	ARC-C acc ↑	WinoG acc ↑	OBQA acc ↑	TQA mc2 acc ↑	Avg acc ↑
AMOR-Mamba2 ♥	29.3	31.4	65.9	55.4	24.7	52.4	21.0	36.4	39.6
AMOR-serial-Mamba2	26.8	31.7	65.6	55.3	22.3	50.2	21.2	36.3	38.7
AMOR-fused-Mamba2	27.1	31.8	66.2	56.2	23.1	53.1	19.2	40.1	39.6
AMOR-Gated DeltaNet ♥	30.2	31.8	65.0	56.0	24.6	51.3	23.8	40.9	40.5
AMOR-serial-Gated DeltaNet	27.7	31.8	65.7	55.8	24.0	49.5	22.8	41.9	39.9
AMOR-fused-Gated DeltaNet	28.0	31.9	65.6	56.8	24.1	51.3	20.4	38.8	39.6

I LIMITATIONS

We fix $\kappa = 0.2$ across scales and backbones, motivated by an earlier small-budget 180M sweep under a fixed-offset rule. That sweep suggested firing on slightly fewer than half of positions, but does not establish a generalizable rule beyond the configurations studied here. The gate also operates on a progressively narrower entropy distribution: the AMOR-Mamba2 Block 0 median rises from 0.706 to 0.934 and its SD falls from 0.106 to 0.028 between 180M and 1.5B, with similar values for AMOR-Gated DeltaNet (Table 7). Whether entropy calibration improves routing in this regime remains open.

Entropy measures uncertainty in the predictive distribution, not correctness or recurrent-state capacity; a confident error can leave the gate closed. Runtime conclusions are specific to the measured kernels, hardware, batch sizes and workloads.