When to Think Fast and Slow?
AMOR: Adaptive Entropy Gate for Hybrid Models

Haoran Zheng
The University of Chicago
haoranzheng@uchicago.edu

Chen Shani
Stanford University
cshani@stanford.edu

Abstract

Recurrent-attention hybrids aim to combine the efficiency of recurrence with the expressivity 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. This yields a simple, gradient-free routing mechanism inspired by uncertainty-driven computation and the System 1 / System 2 distinction.

Across Mamba2 and Gated DeltaNet backbones (180M–1.5B), AMOR consistently matches or outperforms both pure recurrent models and fixed-schedule hybrid baselines while invoking attention on only $\sim$22% of tokens. It achieves strong performance on common-sense reasoning benchmarks and maintains stable long-context performance on LongBench, where prior hybrid models 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 both efficiency and robustness, offering a simple alternative to uniform or fixed routing strategies and pointing toward adaptive hybrid architectures that dynamically match computation to input difficulty.

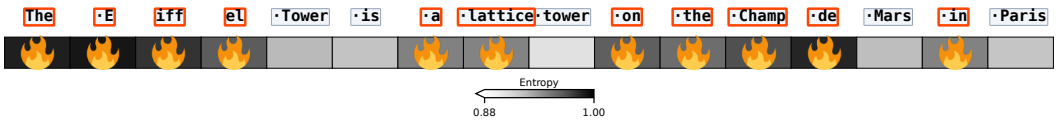

Figure 1: Entropy gate fire pattern of AMOR on the input sentence "The Eiffel Tower is a lattice tower on the Champ de Mars in Paris." The example illustrates how AMOR combines the strengths of state-space and attention mechanisms: the gate fires early in the sequence, when uncertainty over possible 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 state backbone for easier continuations.

Preprint.

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 like Mamba [Gu and Dao, 2023, Dao and Gu, 2024, Lahoti et al., 2026] and Gated DeltaNet [Yang et al., 2025] 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. However, this efficiency comes with a structural limitation. **A finite state cannot faithfully store an unbounded set of distinct 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 Qwen3.5 [Qwen Team, 2026] and Nemotron-3-Super [Chandiramani et al., 2026] are among the strongest open-weight systems, and are competitive with closed-source state-of-the-art models on standard benchmarks. However, current designs still rely on fixed computation layouts, 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 state mechanisms by selectively allocating expensive computation**. We propose AMOR (Adaptive Metacognitive Output Router; see Figure 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 fixed recurrent backbone and appends K entropy-gated attention blocks. Before each block is executed, 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.

Unlike prior approaches, AMOR applies hard, entropy-based gating post-hoc over a recurrent backbone using only the model's own predictive uncertainty. This yields an intuitive, effective, and stable mechanism that does not require any learned routers or calibration procedures.

We pretrain eight architectures at three scales (180M, 440M, and 1.5B parameters) under a uniform Chinchilla-optimal token budget. Across scales, we find that AMOR deploys attention at only roughly 20-30% of greedy-decode positions, suggesting that full attention is unnecessary for most tokens. Despite this sparse attention usage, AMOR achieves the strongest common-sense reasoning performance at every scale (Table 1), improves substantially over its backbone on retrieval benchmarks while remaining competitive with full hybrid architectures (Table 2), and preserves the backbone's robustness on LongBench where both Transformers and serial hybrids degrade significantly (Table 3). Crucially, because attention is routed only to uncertain positions, AMOR maintains the strengths of its underlying state-space backbone rather than overwriting them with indiscriminate attention computation. Our results suggest that efficiency and expressivity are not opposing forces to be traded off, but rather a routing problem of determining **where and when compute should be allocated**.[3]

2 Related Work

We review prior work on recurrent backbones, recurrent-attention hybrids, and conditional compute.

Recurrent backbones. Linear recurrent models have emerged as efficient alternatives to attention, including S4 [Gu et al., 2022], Mamba [Gu and Dao, 2023], Mamba2 [Dao and Gu, 2024],

[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" ([Kahneman, 2011]; page 22).

[3] We release public training, evaluation, and inference benchmarks across all eight architectures (Appendix B); code at `https://github.com/HaoranZhengRaul/AMOR`.

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 sequence mixer can serve as the unmodified backbone before the entropy gates. We study Mamba2 [Dao and Gu, 2024] and Gated DeltaNet [Yang et al., 2025].

State-space backbones achieve efficient inference by compressing past context into a fixed-size recurrent state, enabling linear-time training and constant-time decoding [Dao and Gu, 2024, Yang et al., 2025]. While state models are much faster at inference than any variation of attention ($O(1)$ versus $O(N^2)$), their fixed-size latent state imposes a hard information bottleneck: they cannot reliably store and retrieve arbitrary long-range dependencies when the required signal exceeds the capacity of the state. Empirically, Jelassi et al. [2024] show that fixed-state models 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 state models offer compelling efficiency gains, they struggle to match the flexible, content-addressable memory afforded by attention.**

Recurrent-attention hybrids. To address the limitations of both Transformers and state models, many explored hybrid architectures. We refer to the two dominant classes as *serial* and *fused* hybrids.

Serial hybrids replace the recurrent mixer with attention at a *fixed subset* of layers. Mamba-based hybrids typically use very sparse attention: Jamba [Lieber et al., 2025] pairs a 1:7 attention-to-Mamba ratio (with MoE on top), while Bamba [Chu et al., 2024] and Nemotron-3-Super [Chandiramani et al., 2026] sit near 1:10. Gated DeltaNet hybrids tend toward denser attention: Qwen3.5 [Qwen Team, 2026], Kimi Linear [Kimi Team, 2025], and OLMo Hybrid [Merrill et al., 2026] converge near a 3:1 Gated DeltaNet-to-attention ratio. Other examples include Samba [Ren et al., 2025], Griffin [De et al., 2024], Zamba 2 [Glorioso et al., 2024], Granite 4.0 [IBM, 2025], and Jet-Nemotron [Gu et al., 2025].

Fused hybrids run attention *in parallel* with the recurrent mixer within the same layer. For example, Hymba [Dong et al., 2025] combines Mamba with multi-head causal and sliding-window attention through learned per-channel blending, while Falcon-H1 [Zuo et al., 2025] fuses Mamba2 with full causal attention via flexible per-layer channel allocation. Prior work on hybrid architectures has therefore focused primarily on the *depth* axis: determining where attention should be placed and in what proportion. Bae et al. [2025] show that fused hybrids perform best when attention is distributed throughout the network rather than concentrated near the beginning or end, and that serial hybrids favor an attention-to-Mamba ratio of roughly $1{:}5$. Similarly, Wang et al. [2025] find that linear-to-full-attention ratios between $3{:}1$ and $6{:}1$ recover transformer-level recall. To the best of our knowledge, no fused hybrid uses Gated DeltaNet as the underlying recurrent model.

In contrast, both serial and fused hybrids fix attention placement at design time, whereas AMOR introduces an orthogonal axis: K post-hoc attention blocks appended after a fully executed recurrent backbone, with attention selectively activated per token via normalized-entropy gating.

Conditional Compute. Conditional-compute methods allocate compute along two axes: *how decisions are made* (soft vs. hard gating) and *what is gated* (granularity).

Soft methods such as ACT [Graves, 2016] and PonderNet [Banino et al., 2021] use continuous halting probabilities to mix intermediate outputs, trading extra compute for smoother optimization. Hard methods instead make discrete routing decisions. Mixture-of-Depths (MoD) [Raposo et al., 2024] skips entire transformer blocks per token via a learned router, while sparse attention methods like DeepSeek Sparse Attention [DeepSeek-AI, 2025], Native Sparse Attention [Yuan et al., 2025], and MoBA [Lu et al., 2025] prune attention via top-k masks to reduce quadratic cost. AMOR also belongs to the hard family, applying a per-position binary mask to attention sublayers, but without a learned router or gradient estimator [Bengio et al., 2013].

Prior work typically conditions on the *input* (e.g., router-based or key-similarity signals), while a smaller line of work conditions on the *output*, using the predictive distribution after the `lm_head`. CALM [Schuster et al., 2022] is the main example, using the top-1 vs. top-2 softmax margin with Learn-then-Test thresholding to meet a target risk and performing layer-wise early exiting. In contrast,

AMOR uses normalized entropy over the full distribution with a simple EMA-based adaptive threshold, and applies gating per position within attention sublayers rather than halting layers.

Overall, prior methods either fix attention placement (hybrids), route compute via learned modules, or gate using softmax-based criteria. AMOR combines these ideas via post-hoc attention blocks over a recurrent backbone, gated per token by output entropy without a router.

3 AMOR

We now describe AMOR in detail, starting with the general architecture (§ 3.1), and focusing on our entropy gate (§ 3.2) and each block's flow (§ 3.3) during training (§ 3.4) and inference (§ 3.5).

3.1 Architecture

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

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

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).

Both models share a single LM head, weight-tied to the embedding, which is invoked $K{+}1$ times per forward pass: once after each AMOR block (on its normalized input) to compute the entropy gate (§3.2), and once on the final normalized residual to produce the output logits.

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., 2024] applied to $\mathbf{Q}$ and $\mathbf{K}$. For model dimension D, each block introduces $\sim 4D^2$ parameters ($\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$ plus a per-channel gating vector $\boldsymbol{\alpha} \in \mathbb{R}^D$).

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 before computing the entropy gate and attention, then updates the (unnormalized) residual. This distinction is necessary because the residual is no longer unit-RMS 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 §1. We 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 F). This variant consistently underperforms in both evaluation accuracy and retrieval, indicating that preserving the backbone's LM-ready semantics is important for effective gating and refinement.

Lastly, each AMOR block adds a per-channel $\boldsymbol{\alpha}$-scaled attention update to the residual. The learned scaling vector $\boldsymbol{\alpha} \in \mathbb{R}^D$ 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 bitwise identical to the recurrent backbone at initialization; deviations are learned progressively as $\mathbf{W}_O$ updates. Finally, AMOR blocks contain no MLP (Appendix F), 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. Given the block's normed 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:

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

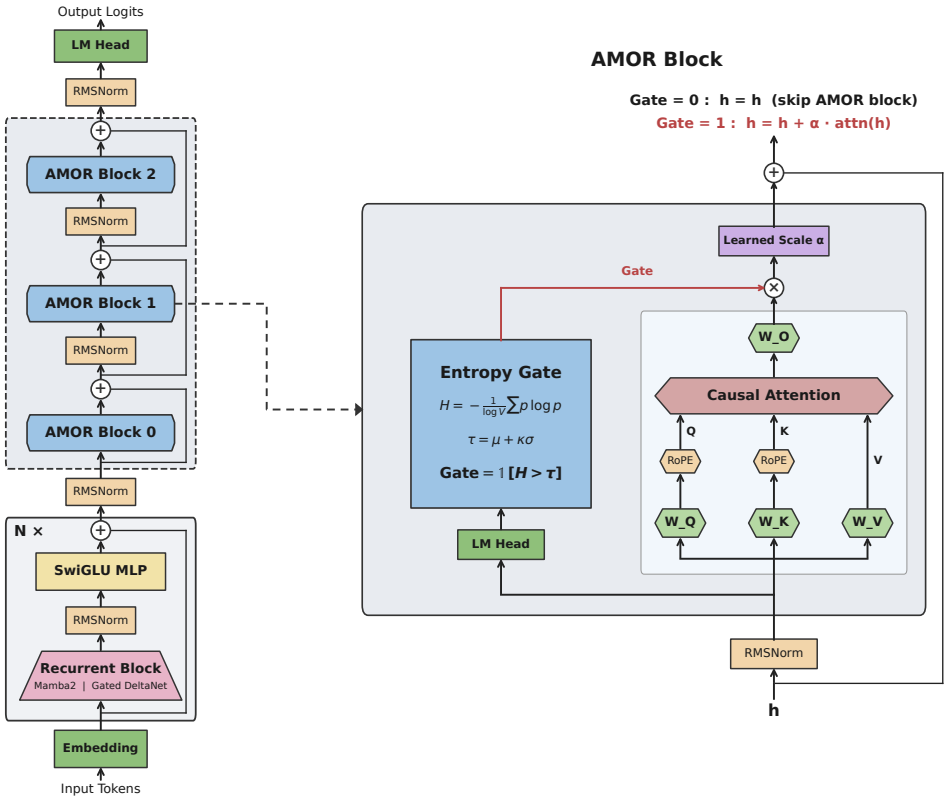

Figure 2: AMOR architecture. A recurrent backbone (Mamba2 or Gated DeltaNet, with SwiGLU MLPs) feeds three AMOR blocks. Each block queries the tied LM head, gates causal RoPE attention via normalized entropy, and adds a per-channel $\boldsymbol{\alpha}$-scaled residual update. Block 0 consumes the backbone's 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.

Logits are detached before entropy computation: $\mathcal{H}_t$ drives the gating decision only and does not backpropagate into the backbone, keeping the gate purely diagnostic. At each training step, the block updates two non-learnable EMA buffers,

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

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

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

The gate is hard-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 C).

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 $\ell \geq 1$, $\tilde{h}_t^{(\ell)} = \mathrm{RMSNorm}_\ell\left(h_t^{(\ell)}\right)$, with

$h_t^{(\ell)}$ the unnormalized residual from block $\ell-1$. Training uses dense attention with a post-mask:

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

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

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

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

We parameterize the per-channel scaling 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. The gate is hard-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.

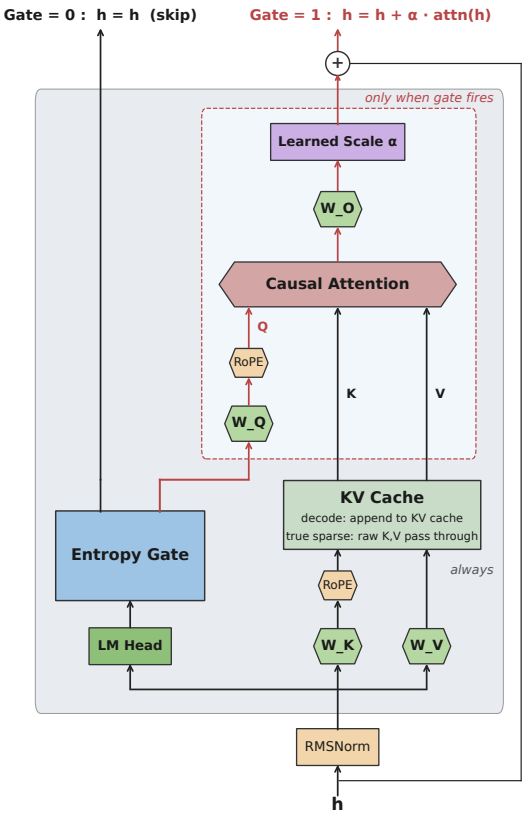

Figure 3: AMOR decode path. When g_t=0, the block skips $\mathbf{W}_Q$, RoPE on $\mathbf{Q}$, SDPA, and $\mathbf{W}_O$; $\mathbf{W}_K$, $\mathbf{W}_V$, RoPE on $\mathbf{K}$, and KV-cache updates always run.

3.4 Training mode

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

3.5 Inference: conditional skip and KV continuity

At decode time, the gate is evaluated per token and induces a conditional skip: $g_t = 0$, 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**. In practice, firing rates increase with sampling temperature and with prompt length beyond the training regime, reflecting broader predictive distributions and reduced backbone certainty (Table 7).

The conditional skip provides a decode-time speedup when the attention compute saved by non-firing tokens outweighs the fixed cost of evaluating the LM head to obtain $\mathcal{H}$:

$$(1-p) \cdot C_{\text{attn}}(L) > C_{\text{lm\_head}}, \tag{9}$$

where p is the firing rate, $C_{\text{attn}}(L)$ is the per-token attention cost at cache length L, and $C_{\text{lm\_head}}$ is the LM-head cost. Speedups therefore require a non-trivial gated-off fraction $(1-p)$ and sufficiently large L for attention to dominate. We derive the crossover in Appendix E and validate it in Figure 5.

4 Experiments

We evaluate AMOR against state-of-the-art sequence models to isolate a single question: *when should a model allocate attention?*

Baseline Architectures. We compare eight architectures spanning three families that share the same computational skeleton (Eq. 1) and differ *only* in how attention is allocated. This controlled design ensures that any performance differences directly reflect the placement-of-attention decision, rather than confounding implementation details. **Backbone models** include pure recurrent mixers (Mamba2 [Dao and 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., 2024]). **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., 2025]. 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. We follow the high-level designs of Jamba and Hymba but remove orthogonal components to isolate the effect of *fixed-schedule attention*. All hybrids use full causal attention (no sliding-window variants) and the same Mamba2 backbone, ensuring consistent attention and recurrent primitives across models. We also exclude unrelated additions such as Hymba's meta tokens and KV reuse, and Jamba's MoE head.

All models are trained from scratch on FineWeb-Edu [Penedo et al., 2024] with the Llama-3.1 tokenizer ($V{=}128{,}256$) at sequence length $T{=}3072$, using an identical optimization recipe (Table 5, Appendix B). At each scale, the token budget follows the Chinchilla-optimal allocation [Hoffmann et al., 2022], keyed to the largest model, so all models see the same data volume: 3.69B / 8.97B / 30.7B tokens at 180M / 440M / 1.5B. Data ordering is fixed across all models and scales.

To ensure comparability, all eight models share the same SwiGLU MLP and scaling configurations: $d_{\text{model}}/N/d_{\text{ff}}{=}768/12/1216$ (180M), $1024/24/1984$ (440M), and $2048/24/4096$ (1.5B). AMOR blocks themselves contain no MLP (Appendix F). Unless otherwise stated, AMOR uses $K{=}3$ gated attention blocks. A $K{=}1$ ablation across both backbones and all scales is reported in Table 12 (Appendix F). A full parameter breakdown per model is provided in Table 4 (Appendix A).

4.1 Results

Following the evaluation protocol of Yang et al. [2025], we evaluate on four settings: (a) common-sense reasoning at 180M, 440M, and 1.5B (§4.1.1); (b) in-context retrieval at 1.5B (§4.1.2); (c) long-context behavior at 1.5B (§4.1.3); and (d) inference efficiency (§4.1.4). Per-task details are in Appendix D. All evaluations use the LM Evaluation Harness [Biderman et al., 2024].

4.1.1 Common-Sense Reasoning

AMOR achieves the highest average across all scales (Table 1), matching or exceeding both pure recurrent backbones and fixed-schedule hybrids that allocate the same number of attention layers at the same head dimension. AMOR-Mamba2 leads at 180M and 1.5B by a clear margin, while at 440M AMOR-Gated DeltaNet leads within a tight band. The 180M results and a depth ablation ($K{=}3$ vs. $K{=}1$ across both backbones and all scales) are reported in Appendix F.

4.1.2 In-Context Retrieval

Cloze retrieval is the regime where attention is least substitutable: each task ends mid-passage, and the model must recover a span from earlier in the input. Pure recurrent backbones therefore underperform sharply, while fixed-schedule hybrids, which allocate attention uniformly, take the lead (Table 2). **Both AMOR variants substantially improve over their recurrent backbones and remain competitive with fixed-schedule hybrids**, while activating attention at only $\sim 22\%$ of decoding steps (Table 7, Appendix E). The same pattern holds on S-NIAH. Beyond the 3072 training context, vanilla RoPE attention degrades out-of-distribution, most notably for pure Transformer.

Table 1: Common-sense reasoning across three scales.

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	402.6	18.3	27.2	60.2	47.6	18.4	50.3	15.6	45.0	35.3
Mamba2	32.47	352.3	16.8	27.4	60.0	45.7	18.5	48.9	**17.2**	43.8	34.8
Gated DeltaNet	31.17	392.6	15.6	27.5	60.9	47.4	19.0	52.1	14.2	43.6	35.0
Mamba2 Serial Hybrid	30.69	398.9	18.8	27.5	60.6	46.8	**19.7**	**53.0**	15.4	43.0	35.6
Gated DeltaNet Serial Hybrid	**30.24**	366.7	18.1	27.3	**61.6**	45.7	18.6	52.6	15.8	41.3	35.1
Mamba2 Fused Hybrid	30.73	304.3	18.7	**27.6**	60.1	45.5	18.6	49.4	16.0	44.0	35.0
AMOR-Mamba2 ♥	31.46	240.3	**20.3**	27.4	60.5	46.5	18.0	51.3	16.6	**45.9**	**35.8**
AMOR-Gated DeltaNet ♥	30.60	**234.2**	19.5	27.3	61.0	**47.7**	17.3	51.1	16.2	45.2	35.7
440M											
Transformer	19.92	74.1	27.8	31.2	64.9	55.9	23.9	49.9	20.4	**42.4**	39.5
Mamba2	19.78	74.2	25.3	31.9	65.2	**56.9**	23.2	51.1	21.6	36.7	39.0
Gated DeltaNet	19.39	62.4	26.6	31.9	**66.3**	56.7	23.4	50.5	21.2	40.7	39.7
Mamba2 Serial Hybrid	19.11	69.3	28.6	**32.0**	66.1	55.9	23.6	50.8	20.4	38.8	39.5
Gated DeltaNet Serial Hybrid	**18.94**	56.2	29.6	31.8	65.8	**56.9**	24.3	52.0	21.6	39.5	40.2
Mamba2 Fused Hybrid	19.24	58.2	28.9	31.9	65.8	56.1	24.3	51.9	22.2	**42.4**	40.4
AMOR-Mamba2 ♥	19.60	57.4	29.3	31.4	65.9	55.4	**24.7**	**52.4**	21.0	36.4	39.6
AMOR-Gated DeltaNet ♥	19.27	**50.2**	**30.2**	31.8	65.0	56.0	24.6	51.3	**23.8**	40.9	**40.5**
1.5B											
Transformer	13.41	25.9	38.2	38.3	69.5	65.2	29.9	52.2	23.8	35.4	44.1
Mamba2	13.32	26.7	36.1	38.9	70.3	66.1	30.4	**55.1**	25.6	34.3	44.6
Gated DeltaNet	13.11	22.0	38.5	39.2	70.7	67.7	**33.4**	53.1	25.0	35.0	45.3
Mamba2 Serial Hybrid	12.97	25.3	38.1	**39.4**	**70.9**	**67.9**	31.3	53.0	26.0	35.4	45.3
Gated DeltaNet Serial Hybrid	**12.94**	22.3	**39.7**	39.2	70.8	64.9	30.5	54.2	24.6	37.7	45.2
Mamba2 Fused Hybrid	13.04	23.7	37.5	39.2	70.4	67.4	32.3	53.7	26.6	35.2	45.3
AMOR-Mamba2 ♥	13.30	23.0	39.2	38.6	**70.9**	67.0	30.2	54.7	**26.8**	**39.0**	**45.8**
AMOR-Gated DeltaNet ♥	13.10	**21.5**	39.1	39.2	70.6	66.4	30.5	53.7	24.8	36.9	45.1

Table 2: In-context retrieval and S-NIAH at 1.5B.

Model	SWDE	SQuAD	FDA	TQA	NQ	DROP	NIAH-Single-1			NIAH-Single-2			NIAH-Single-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: Accuracy on 14 tasks from LongBench [Bai et al., 2024] at 1.5B: NarrativeQA, Qasper, MultiFieldQA-en, HotpotQA, 2WikiMultihopQA, MuSiQue, GovReport, QMSum, MultiNews, TREC, TriviaQA, SAMSum, LCC, and RepoBench-P by order.

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	1.6	4.9	0.6	1.4	0.2	5.3	5.8	10.7	3.0	3.3	3.6	9.0	7.7	4.1
Mamba2	1.6	3.9	11.1	**4.6**	6.8	**2.8**	7.5	**15.6**	11.3	7.0	14.3	7.2	10.1	11.0	8.2
Gated DeltaNet	1.4	3.6	9.4	3.3	**8.3**	2.4	8.5	14.9	10.8	5.5	**17.0**	**19.3**	9.5	10.8	**8.9**
Mamba2 Serial Hybrid	0.2	2.2	4.4	0.2	2.5	0.4	2.8	4.5	10.0	4.5	4.2	6.6	10.0	10.3	4.5
Gated DeltaNet Serial Hybrid	0.5	2.4	5.6	1.3	4.8	1.1	4.1	5.6	10.5	7.5	9.2	5.4	9.6	11.2	5.6
Mamba2 Fused Hybrid	1.2	3.4	9.5	3.1	6.4	2.2	**9.3**	14.8	8.1	3.0	14.2	6.0	**11.2**	10.8	7.4
AMOR-Mamba2 ♥	1.6	3.3	10.1	2.9	6.1	1.8	7.6	14.7	**13.2**	**13.0**	13.2	7.5	9.3	**11.5**	8.3
AMOR-Gated DeltaNet ♥	**1.7**	**4.0**	**11.4**	3.6	7.0	1.9	8.1	15.1	10.8	11.0	11.8	11.4	10.1	10.3	8.4

4.1.3 Long-Context Behavior

LongBench evaluates long-context understanding beyond the training regime. Pure recurrent backbones, which do not explicitly encode position, degrade gracefully, while fixed-schedule hybrids invoke RoPE attention without length extension and fall below the recurrent baseline (Table 3).

AMOR mitigates this RoPE distribution shift via its entropy gate: when the gate is quiet, the attention path is skipped and the recurrent backbone, free of positional drift, carries the prediction; when the gate fires (up to $\sim 75\%$ at 16K prompts; Table 7), the α-blended attention contribution still keeps AMOR above its hybrid counterparts. The same pattern appears in perplexity under length extrapolation, both on NarrativeQA (Fig. 4) and across the full six-benchmark grid (Figure 7, Appendix D.3).

To conclude, **across model sizes and different benchmarks, AMOR maintains the advantage of the backbone while also using attention when needed (when the model is uncertain)**.

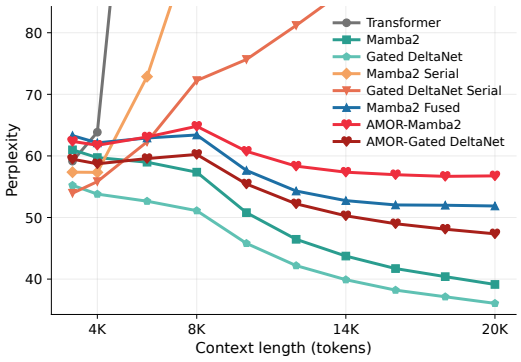

Figure 4: Per-token perplexity on NarrativeQA at 1.5B versus context length. Pure recurrent backbones and AMOR variants stay flat or improve as context grows; fixed-schedule serial hybrids drift upward, and Transformer collapses.

4.1.4 Decode efficiency

We analyze the compute tradeoff introduced by AMOR's conditional attention mechanism. AMOR replaces unconditional attention with entropy-based routing, trading a per-block `lm_head` probe for selectively skipping attention on confident tokens (§3.5). Theoretically, this is increasingly favorable for longer contexts: probe cost is constant in cache length, while attention cost grows linearly (Appendix E). Thus, conditional skipping amortizes routing overhead, and savings scale with both context length and the fraction of skipped positions.

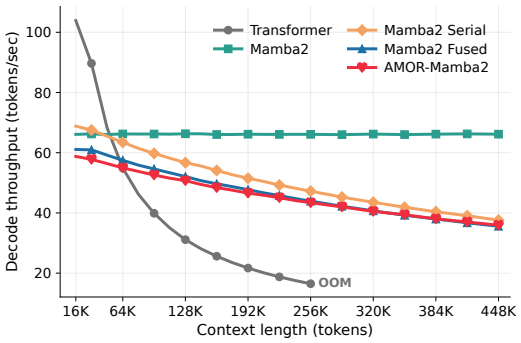

Figure 5: Decode tokens/sec vs context length at 440M (AMOR at $\sim 40\%$ fire rate). AMOR exhibits better decode throughput scaling (less degradation) with longer context. Note: Transformer drops out at 288K due to KV-cache memory wall on H100.

Empirically, this behavior is shown in Fig. 5. Under a simulated 40% gate fire rate[4] on a single H100 NVL, AMOR degrades more gracefully than hybrid baselines as context length increases. Its advantage grows in the long-context regime: AMOR surpasses the Mamba2 Fused Hybrid at ~ 256K context and matches the Mamba2 Serial Hybrid within noise at 448K.

The theoretical crossover analysis and full benchmarking setup with Gated DeltaNet are in Appendix E; detailed decoding and training throughput results are reported in Table 8.

5 Conclusion

AMOR is a post-hoc hybrid that allocates attention based on predictive uncertainty. Across model scales and evaluation settings, AMOR matches or exceeds both pure recurrent backbones and fixed-schedule hybrids, while invoking attention on only a fraction ($\sim 22\%$) of tokens.

Our results suggest that *when* attention is applied matters as much as *how much*: selectively engaging attention based on the model's own uncertainty yields a more efficient and robust allocation than fixed schedules. More broadly, AMOR demonstrates that simple, output-driven mechanisms can effectively coordinate hybrid architectures without additional learned routers or auxiliary objectives, pointing toward a class of **models that adapt their computation to the demands of the model**.

[4]Targeted during training; the trained models achieve $\sim 22\%$, but we use 40% for a conservative estimate.

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.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. URL https://arxiv.org/abs/1308.3432.

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.

Aakshita Chandiramani, Aaron Blakeman, Abdullahi Olaoye, Abhibha Gupta, Abhilash Somasamudramath, Abhinav Khattar, Adeola Adesoba, Adi Renduchintala, Adil Asif, Aditya Agrawal, et al. Nemotron 3 super: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2604.12374*, 2026. URL https://arxiv.org/abs/2604.12374.

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.

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*, 2023. 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)*, pages 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, Jack W. Rae, Oriol Vinyals, 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.

IBM. 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.

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)*, pages 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.

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. In *International Conference on Learning Representations (ICLR) 2025*, 2025. 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.

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.

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.

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*, pages 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.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 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, 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.

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 Architectural Details

The eight architectures share $d_{\text{model}}/N/d_{\text{ff}}$ at each scale (Table 4), where N is the number of mixer-MLP layers (attention/recurrent). The "Attn" column reports where attention is applied; AMOR's $K{=}3$ post-hoc blocks are not counted in N. We fix N across architectures for depth-equivalent comparison. Because mixer types differ in parameterization, this does not equalize total parameter count: multi-head attention layers are naturally lighter than Mamba2 or Gated DeltaNet mixers. The additional AMOR blocks introduce a modest parameter overhead (Table 4), but this increase is small relative to the backbone ($< 5\%$) and does not account for the observed gains.

Table 4: Per-model parameter counts and layer layouts.

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	—	12 / 24 / 24
Gated DeltaNet	174.9M	429.5M	1,473.7M	0	—	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

B Training Recipe

All architectures and ablations are trained under a uniform recipe, using identical hyperparameters within each scale (Table 5); differences across scales are limited to peak learning rate and total step count. At a given scale, all models are trained on the same number of tokens. This token budget is set using the Chinchilla-optimal $20\times$ ratio [Hoffmann et al., 2022] for a reference model (AMOR-Mamba2), and then held fixed across architectures to ensure a controlled comparison independent of parameter count differences.

Table 5: Training recipe summary.

	180M	440M	1.5B
Tokens	3.69B	8.97B	30.7B
Steps	7,503	18,255	62,558
Sequence length	3072	3072	3072
Effective batch (tokens)	491,520	491,520	491,520
Optimizer	AdamW $(\beta_1, \beta_2){=}(0.9, 0.95)$, weight decay 0.1, grad clip 1.0		
Backbone peak LR	$3{\times}10^{-4}$	$3{\times}10^{-4}$	$1.5{\times}10^{-4}$
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	$3{\times}10^{-3}$	$3{\times}10^{-3}$	$3{\times}10^{-3}$
$\boldsymbol{\alpha}$ 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

All weights are re-initialised to $\mathcal{N}(0, 0.02)$ after model construction, overriding library defaults that differ across implementations (HF Mamba2, `flash-linear-attention` Gated DeltaNet layer, and PyTorch `nn.Linear`). This ensures a consistent initialization scheme across architectures and

removes a potential source of variation unrelated to the model design. The only intentional exception is $\mathbf{W}_O$ in every AMOR block, which is initialized to zero so that the block starts as an exact no-op. Embedding and LM head are weight-tied, and all linear layers use `bias=False`.

B.1 Training Mode

We train in 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 true_sparse formulation that computes attention only at firing positions; both modes produce bit-identical residual updates at each position.

We use the dense formulation for training purely for efficiency: it maps to PyTorch's `scaled_dot_product_attention` (FlashAttention-2 kernels), whereas 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.

C Gate Statistics Across Scales

Table 6 reports end-of-training gate statistics for AMOR Block 0 across both backbones and all three scales. We show the EMA-tracked batch-median μ, batch-standard deviation σ, the frozen threshold $\tilde{\tau}{=}\mu + 0.2\sigma$, and the per-channel sigmoid weight $\boldsymbol{\alpha}$. These quantities are aggregated over a $K{=}20$ entropy-log window spanning $\sim$10k post-convergence steps, reflecting steady-state behavior.

To assess lifetime stability, we additionally report the fire-rate 10-90% range over the full training trajectory. Figure 6 complements this with per-block entropy histograms at 1.5B scale, with the frozen τ overlaid (red line), showing that the learned threshold remains well aligned with the entropy distribution after convergence.

Table 6: Gate statistics across scales.

Model	**Scale**	**Val PPL**	μ	σ	$\tilde{\tau}$	$\boldsymbol{\alpha}$	**Fire rate** (10–90%)
AMOR-Mamba2 ♥	180M	31.46	0.706	0.106	0.728	0.801	0.33–0.46
AMOR-Mamba2 ♥	440M	19.60	0.847	0.057	0.858	0.810	0.34–0.48
AMOR-Mamba2 ♥	1.5B	13.30	0.934	0.028	0.940	0.662	0.33–0.49
AMOR-Gated DeltaNet ♥	180M	30.60	0.707	0.108	0.728	0.779	0.33–0.45
AMOR-Gated DeltaNet ♥	440M	19.27	0.853	0.053	0.864	0.613	0.34–0.47
AMOR-Gated DeltaNet ♥	1.5B	13.10	0.937	0.030	0.943	0.544	0.33–0.51

D Evaluation setup

We follow the evaluation convention of Yang et al. [2025]. All evaluations run via the LM Evaluation Harness [Biderman et al., 2024]; loglikelihood tasks use the harness defaults, and generation tasks use greedy decoding.

D.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).

D.2 In-Context Retrieval

In line with the evaluation convention of Yang et al. [2025], we evaluate the 1.5B-scale models using greedy decoding on two task suites. The first is the real-world cloze-completion suite of

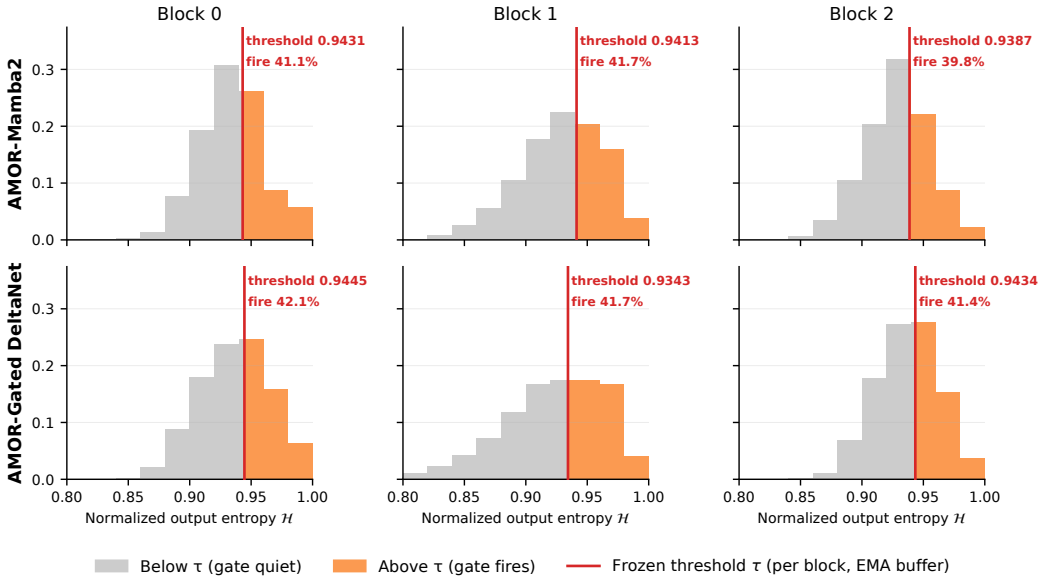

Figure 6: End-of-training entropy distribution per AMOR block at 1.5B; frozen threshold τ as a red vertical line.

Arora et al. [2024a], 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).

D.3 Long-Context Behavior

In line with the evaluation convention of Yang et al. [2025], we evaluate the 1.5B-scale models using greedy decoding on 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 and 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].

D.4 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. 7). 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 (GitHub-Python code), and PG19 [Rae et al., 2020] (long-form public-domain books).

As shown in Fig. 7, RoPE-induced distribution shift beyond 3K leads to sharp perplexity increases for the Transformer baseline and the two serial hybrid models. AMOR does not suffer

Table 7: AMOR decode fire rate vs prompt length and temperature; same frozen $\tilde{\tau}$ for every column.

Model	Fire @ T=0			Fire @ T=1		
	3K	8K	16K	3K	8K	16K
AMOR-Mamba2 ♥	0.208	0.220	0.754	0.242	0.335	0.927
AMOR-Gated DeltaNet ♥	0.301	0.186	0.638	0.194	0.324	0.821

as much, behaving more similarly to state models. For visualization clarity, the y-axes in each panel are linearly capped at 1.3× 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.

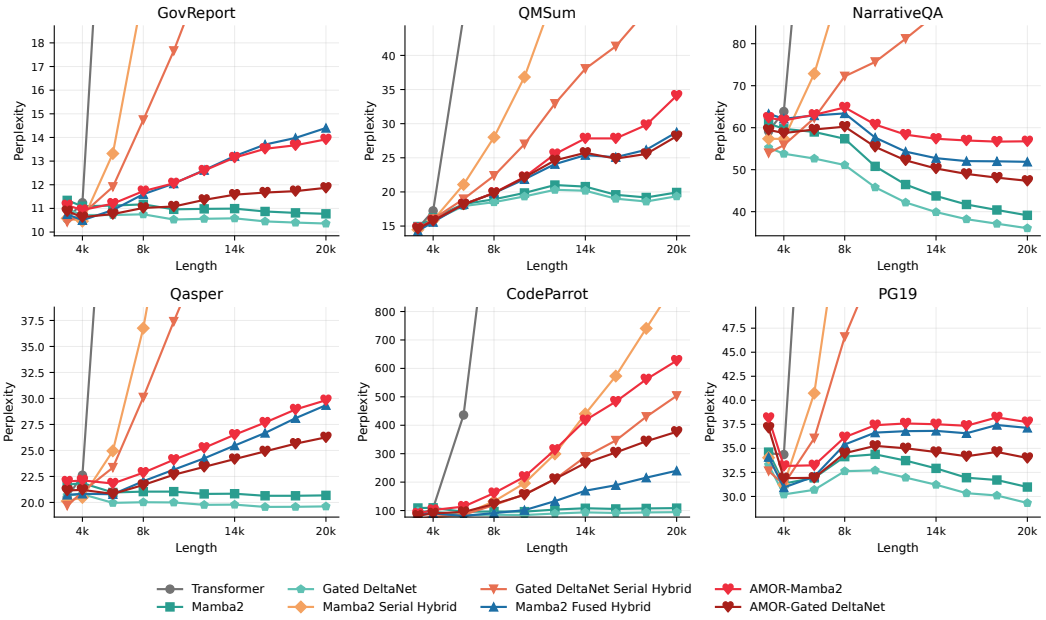

Figure 7: Length-extrapolation perplexity at 1.5B on six long-context benchmarks.

E Decode Efficiency

We benchmark autoregressive decode on a single H100 NVL GPU, timed over 3 runs after a 32-token warm-up; reported numbers are means. Decode uses greedy (T=0) and sampling (T=1) policies on prompts drawn from real text. Fire rates across prompt lengths and temperatures appear in Table 7; full train / decode throughput at 1.5B is in Table 8.

Table 8: Train and decode throughput at 1.5B (single H100 NVL, 3K prompt = training context).

Model	Train tok/s	Decode tok/s	
		T=0@3K	T=1@3K
Transformer	23.6k	102.0	100.5
Mamba2	25.1k	63.8	64.6
Gated DeltaNet	22.8k	44.6	44.4
Mamba2 Serial Hybrid	24.6k	64.5	64.7
Gated DeltaNet Serial Hybrid	23.4k	48.6	48.1
Mamba2 Fused Hybrid	15.3k	60.1	59.8
AMOR-Mamba2 ♥	21.5k	57.8	57.9
AMOR-Gated DeltaNet ♥	19.8k	41.7	41.9

E.1 Decode Break-Even

Eq. 9 expresses the condition under which AMOR's gated decode amortizes its per-block `lm_head` probe overhead with the per-position attention compute saved on non-firing positions. Both costs are per-token and per-block, so the inequality $(1-f)\, C_{\text{attn}}(L) > C_{\text{lm\_head}}$ yields a single breakeven $L_{\text{breakeven}}$ in cached prompt length. At decode a single new query (dimension D) attends to a KV cache of length L:

- $\mathbf{QK}^\top$: a $(1, D)\times(D, L)$ matrix multiply, $2DL$ FLOPs (one multiply and one add per output entry).
- $\mathbf{AV}$: a $(1, L)\times(L, D)$ matrix multiply, $2LD$ FLOPs.
- Softmax: $O(L)$, negligible.
- $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$ projections: $O(D^2)$ each, do not scale with L, drop out of the long-context comparison.

The LM head is a single $D \to V$ linear: $C_{\text{lm\_head}} = 2VD$ FLOPs per token. Multi-head attention partitions the two matmuls across H heads of dimension D/H and sums their contributions, giving the same $4LD$ total. Solving:

$$\begin{aligned} C_{\text{attn}}(L) &= 4LD, \\ C_{\text{lm\_head}} &= 2VD, \\ (1-f)\cdot 4LD &= 2VD \qquad \text{(Eq. 9 at equality)} \\ L_{\text{breakeven}} &= \frac{V}{2(1-f)}. \end{aligned}$$

We note that empirical results might have a longer crossover point than the closed-form derivation, due to decode at batch 1 being bandwidth-bound, which compresses any FLOP-derived savings into smaller wall-time savings, and different attention-layer placements in hybrid designs.

E.2 Empirical Crossover Testing

We bench Fig. 5's five 440M-scale lines on a single H100 NVL, batch 1, bfloat16 autocast: Transformer, Mamba2, Mamba2 Serial Hybrid, Mamba2 Fused Hybrid, and AMOR-Mamba2; the Gated DeltaNet Fused Hybrid is omitted due to the lack of an open-weight implementation. Each length $L \in \{16, 32, 48, \ldots, 256, 288, 320, \ldots, 448\}$K runs three trials of 32 greedy-decode tokens after 8 warm-up tokens on a real FineWeb-Edu prompt; reported numbers are trial means. AMOR's gate is simulated at fire rate $f \approx 0.4$ to match the training target (§3.2); the per-block entropy `lm_head` probe still runs and is included in the timing. To reach longer contexts within a single H100 NVL's 94 GB memory, the bench slices the hidden state to its last position before the final $\mathrm{lm\_head}$ matmul, sparing the $L\times V \approx 66$ GB logit tensor from prefill peak memory; only the last-position logit ever seeds the next token. Applied uniformly to all five models for equal footing, the patch fires only on multi-position inputs and runs after every $\mathbf{W}_K, \mathbf{W}_V$ projection, leaving the decode-time cache state bit-identical to a natural prefill.

Fig. 8 extends the same setup with the three Gated DeltaNet-backbone models (Gated DeltaNet, Gated DeltaNet Serial Hybrid, AMOR-Gated DeltaNet).

F Ablations

G Additional Gate-Fire Pattern Examples

Figure 9 shows gate-fire traces on four prompts covering different domains: a DNA-sequence excerpt, a mathematical-reasoning prompt, a news lead, and a poetry fragment.

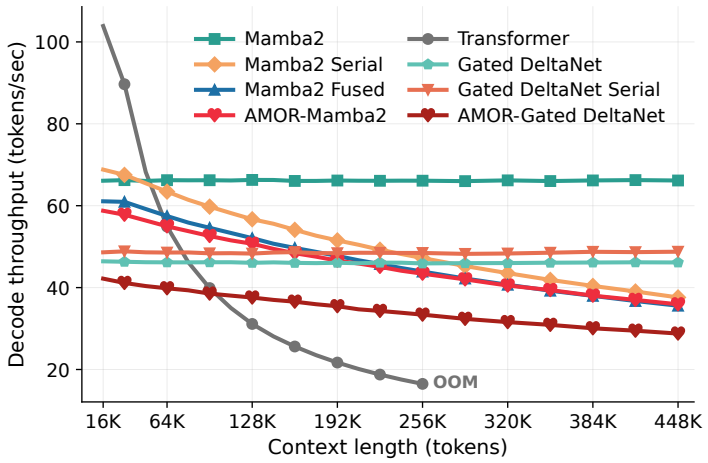

Figure 8: Decode tokens/sec vs context length at 440M for all eight architectures (AMOR variants at $\sim 40\%$ fire rate).

Table 9: 440M ablations on Common-sense reasoning.

Model	LAMB acc ↑	HSwag acc ↑	PIQA acc ↑	ARC-E acc ↑	ARC-C acc ↑	WinoG acc ↑	OBQA acc ↑	TQA mc2 acc ↑	Avg acc ↑
AMOR-Mamba2 ♥	**29.32**	31.42	65.94	55.43	**24.74**	**52.41**	21.00	36.37	**39.58**
AMOR (Block 0 pre-norm) Mamba2	28.31	31.67	65.07	56.48	22.10	49.17	**23.00**	35.75	38.94
Mamba2 Post-hoc Hybrid No Gate	28.90	31.53	66.00	55.35	22.87	47.59	21.60	37.07	38.86
AMOR-MLP-Mamba2	27.38	**31.78**	**66.21**	**56.69**	23.12	51.07	19.20	**39.16**	39.33
AMOR-Gated DeltaNet ♥	**30.22**	31.82	65.02	56.02	24.57	**51.30**	**23.80**	40.87	**40.45**
AMOR (Block 0 pre-norm) Gated DeltaNet	27.94	31.74	65.02	**56.94**	**25.09**	50.51	20.00	**42.13**	39.92
Gated DeltaNet Post-hoc Hybrid No Gate	28.64	31.81	65.83	56.36	24.66	49.57	20.00	39.44	39.54
AMOR-MLP-Gated DeltaNet	27.44	**31.86**	**66.00**	55.85	24.57	50.99	20.20	39.28	39.52

Table 10: 1.5B common-sense ablation.

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	23.0	39.2	38.6	**70.9**	**67.0**	30.2	**54.7**	**26.8**	**39.0**	**45.8**
AMOR (Block 0 pre-norm) Mamba2	**13.25**	**20.8**	**40.1**	**38.9**	70.6	66.0	**31.8**	53.0	24.6	38.5	45.4
AMOR-Gated DeltaNet ♥	13.10	**21.5**	**39.1**	**39.2**	70.6	66.4	30.5	53.7	**24.8**	**36.9**	45.1
AMOR (Block 0 pre-norm) Gated DeltaNet	**13.09**	22.6	38.2	39.1	**71.4**	**67.4**	**32.3**	**55.4**	24.4	35.5	**45.5**

Table 11: 1.5B retrieval and S-NIAH ablation.

Model	SWDE	SQuAD	FDA	TQA	NQ	DROP	NIAH-Single-1			NIAH-Single-2			NIAH-Single-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 (Block 0 pre-norm) 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 (Block 0 pre-norm) 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 12: AMOR block depth $K{=}3$ vs $K{=}1$ (-CLASSIC) on Common-sense reasoning.

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 ♥ ($K{=}3$)	**20.34**	27.38	60.50	46.55	18.00	51.30	16.60	**45.90**	35.82
AMOR-Classic-Mamba2 ($K{=}1$)	19.70	**27.45**	**60.66**	**47.56**	**18.69**	**51.85**	**18.40**	44.35	**36.08**
AMOR-Gated DeltaNet ♥ ($K{=}3$)	**19.54**	27.33	**60.99**	**47.69**	17.32	51.07	16.20	**45.24**	**35.67**
AMOR-Classic-Gated DeltaNet ($K{=}1$)	18.96	**27.39**	60.12	46.68	**19.03**	**51.30**	**16.60**	44.13	35.53
440M									
AMOR-Mamba2 ♥ ($K{=}3$)	**29.32**	31.42	**65.94**	55.43	**24.74**	**52.41**	21.00	36.37	39.58
AMOR-Classic-Mamba2 ($K{=}1$)	29.21	**31.57**	65.61	**56.19**	23.89	50.43	**21.60**	**39.48**	**39.75**
AMOR-Gated DeltaNet ♥ ($K{=}3$)	**30.22**	31.82	65.02	56.02	**24.57**	**51.30**	**23.80**	40.87	**40.45**
AMOR-Classic-Gated DeltaNet ($K{=}1$)	28.39	**31.95**	**65.72**	**56.90**	24.32	49.64	21.80	**41.67**	40.05
1.5B									
AMOR-Mamba2 ♥ ($K{=}3$)	**39.20**	38.58	**70.95**	67.05	30.20	**54.70**	**26.80**	**39.03**	**45.81**
AMOR-Classic-Mamba2 ($K{=}1$)	36.70	**38.94**	70.24	**67.55**	**30.63**	53.35	23.80	34.63	44.48
AMOR-Gated DeltaNet ♥ ($K{=}3$)	**39.06**	39.23	70.57	**66.37**	**30.46**	53.67	24.80	**36.89**	**45.13**
AMOR-Classic-Gated DeltaNet ($K{=}1$)	38.44	**39.43**	**71.00**	65.66	30.12	**53.83**	**26.60**	35.49	45.07

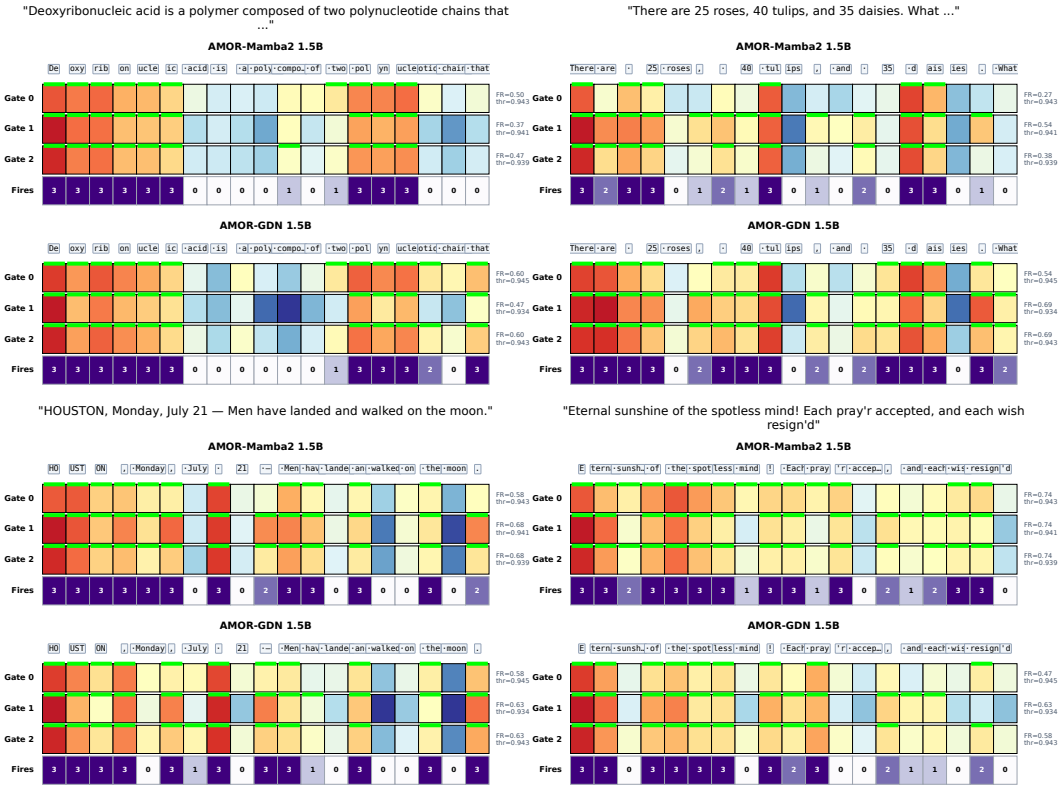

Figure 9: AMOR-Mamba2 1.5B and AMOR-Gated DeltaNet 1.5B gate-fire patterns on four prompts. Same color scheme as Fig. 1.