LEARNING TO COPY, LEARNING TO DEPEND: LANGUAGE MODELS OFFLOAD IN-CONTEXT COPYING TO A POINTER HEAD

Anonymous authors
Paper under double-blind review

ABSTRACT

Many next words are copies: a character's name, a number, a term defined a paragraph earlier. We train language models jointly with a pointer-copy head that copies such tokens from context, then remove it at inference: accuracy on LAMBADA, where most targets appear earlier in the passage, collapses below 7%, far below the 18–23% of baselines trained without it, while eight other benchmarks barely move. We find this output-pathway offloading, in which the backbone's vocabulary readout delegates in-context copying to the head rather than duplicating it, in all five architecture families: two Transformers, Mamba-1, a Jamba-style hybrid, and Gated DeltaNet, each trained from scratch as a controlled baseline/+copy pair at 130M–198M, with further pairs from 45M to 350M. In both Transformers, a linear probe decodes copy targets from the +copy backbone's final hidden state as well as from the baseline's, while its native vocabulary readout becomes less accurate on them. With the head in place, WikiText-103 perplexity and LAMBADA accuracy improve in every pair, and held-out perplexity in all five 5B-token pairs. Copyability stratification confines every pair's LAMBADA gain to targets present in the preceding context, and randomizing the retrieved identities drops every model below its baseline. Under one fixed head configuration, Mamba-3 SISO—a backbone redesigned for retrieval—still benefits on every headline metric. A three-seed replication at 60M reproduces every headline gain's sign and the closed-gate collapse. Because the head configuration differs between architecture classes, we do not rank architectures; all 130M–350M results are single-seed.

1 INTRODUCTION

Pointer-copy mechanisms (Vinyals et al., 2015; Gu et al., 2016; Gulcehre et al., 2016; Merity et al., 2017; See et al., 2017) were once central to neural language modeling, giving a model an explicit retrieval path from context beside learned generation. As Transformers scaled, attention heads learned in-context retrieval implicitly (Olsson et al., 2022) and dedicated copy heads disappeared.

This deserves re-examination. Efficient sequence models have diversified: state space models (SSMs) such as Mamba (Gu & Dao, 2024), SSM–attention hybrids such as Jamba (Lieber et al., 2024), and gated delta-rule networks such as Gated DeltaNet (GDN) (Yang et al., 2025)—together, the *SSM-family* models below—run in linear time but *lack the attention-based retrieval* that made copying redundant. Prior work has measured this recall deficit (Fu et al., 2023; Jelassi et al., 2024; Arora et al., 2024a), and every proposed remedy modifies the backbone (Section 2). We ask a complementary, deliberately narrow question: for a given backbone, with or without native attention, does a *fixed*, output-side copy module help, on what, and by what mechanism? We do not ask which backbone benefits most, which would require one copy configuration on all backbones; ours is windowed and capped on the SSM-family models, for cost and stability, and full-prefix and uncapped on the Transformers (Section 3). The head itself is not new—a pointer-sentinel-style mixture (Merity et al., 2017) with learned query/key projections and a warmed-up, capped gate; the contribution is the controlled five-family study, its inference-time controls, and the costs it exposes.

To answer it, we attach a compact pointer-copy module (394K–1.3M parameters, under 0.9% of the backbone) to five architecture families at 130M–198M scale and train each jointly from scratch on

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

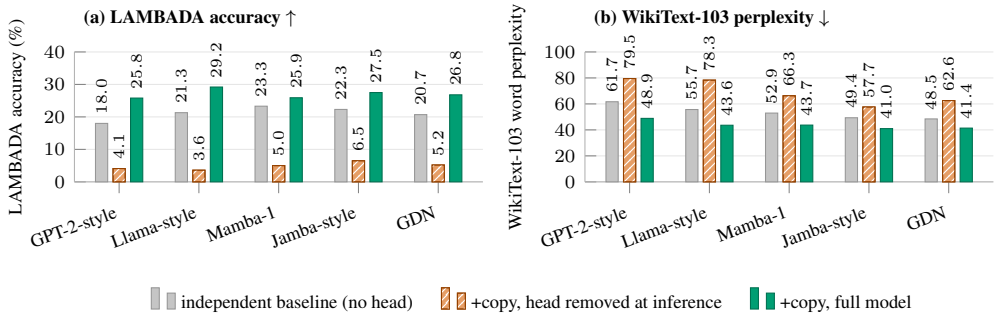

Figure 1: **Copy head gains and closed-gate collapse across five families** (5B tokens, single seed). Gray: baseline; green: +copy; hatched: gate closed. Variant B (groups 1–2) vs. Variant A at c=0.5 (groups 3–5); magnitudes not comparable across that split.

FineWeb-Edu (Penedo et al., 2024) at two token budgets. Each baseline/+copy pair shares data, seed, data order, and optimizer; we add 350M runs for two families, capacity controls at 45M, a three-seed replication at 60M (all four metrics improve in all 15 seed pairs), inference-time identity and closed-gate controls on every 5B model, and a Mamba-1 vs. Mamba-3 SISO (single-input, single-output) comparison under one copy configuration at 2B tokens. Figure 1 shows the 5B outcome on the two headline benchmarks. Our findings, each established within pairs:

- **The copy head helps in every pair.** At 5B tokens, WikiText-103 perplexity falls 9.3–18.2% (14.6–21.7% before excluding the tokenization artifact of Section 5.1), LAMBADA accuracy rises 2.6–7.9 points, and perplexity on a FineWeb-Edu crawl post-dating the training subset falls 3.3–9.9% in all five pairs (Section 5.1); the WikiText gain persists at 350M for two families (Section 5.5). Within the Variant A group, the Jamba-style hybrid, which already contains two full attention layers, gains more LAMBADA accuracy than Mamba-1 (+5.2 vs. +2.6pp), so in-backbone attention at this scale does not make the output-side head redundant (Section 6).
- **The output pathway offloads retrieval to the head.** With the copy pathway removed at inference (the gate closed), LAMBADA accuracy falls to 3.6–6.5% (independent baselines: 18–23%) while eight of nine multiple-choice benchmarks stay within 1.6 points of the full model; on WikiText-103 tokens absent from the prior 512 tokens, the three closed-gate SSM-family backbones are 6% *better* than their baselines (Section 5.4). A linear probe on the two Transformer pairs detects no reduction in how well copyable targets can be decoded from the backbone's final hidden state, while the native vocabulary readout becomes less accurate on them, and the backbone cannot ship without the head: a division of labor, not an add-on.
- **The gain is confined to copyable targets, and it leaks.** Splitting LAMBADA by whether the target occurs in the prior 512 tokens (74% of examples) confines every pair's gain to copyable targets, where accuracy rises 4–11 points; on the rest LAMBADA perplexity worsens in every pair because copy mass lands where nothing is copyable (Section 5.3).
- **The gain requires correct context-token identities.** Replacing each attended source's identity with that of a random earlier position at inference, attention and gate untouched, drops all five models *below* baseline (LAMBADA 2–4% vs. 18–23%; Section 6); at 45M, FLOP- and parameter-matched baselines recover at most 4.0 of the 14.4 WikiText perplexity points the head gains and under 1 of its 14.3 LAMBADA points (Appendix J).
- **The effect on in-context reasoning is pair-dependent.** On a key–value overwrite probe (MQAR-update: retract a superseded value), the head lowers accuracy sharply in one pair (GDN, 82.5→28.0%), by about two standard errors in two more (200 items per task), leaves one flat, and raises it by 22 points in the GPT-2-style pair, whose head configuration differs (a GPT-2-style model retrained with the SSM-family head, Appendix R, was not scored on this probe; Section 6).

- **A backbone redesigned for retrieval still benefits.** Under one fixed Variant A head at 2B tokens, Mamba-3 SISO starts from a stronger baseline than Mamba-1 (23.5% vs. 18.2% LAMBADA, both at 2B) and still gains on every headline metric (WikiText-103 -15.6%, LAMBADA $+2.9$ points at the gate cap used in training, $c = 0.5$), and the gain is identity-specific (Section 5.6).

All 130M–350M results are single-seed; a three-seed replication at 60M reproduces the sign of every headline gain in all five families (Appendix K). Code, evaluation scripts, and all result files are in the supplementary material.

2 Related Work

Pointer networks (Vinyals et al., 2015), CopyNet (Gu et al., 2016), pointer-sentinel mixtures (Gulcehre et al., 2016; Merity et al., 2017), and pointer-generator networks (See et al., 2017) mix a distribution over context tokens with a vocabulary softmax; the continuous neural cache (Grave et al., 2017), an output-side distribution over recent token identities keyed by hidden states, is the closest prior formulation and the one we benchmark against (Appendix U). Ours adds learned query/key projections and a warmed-up, capped gate to that cache, and joint training with non-Transformer backbones to the pointer-sentinel mixture; the head itself is not a contribution.

kNN-LM (Khandelwal et al., 2020), RETRO (Borgeaud et al., 2022), Memorizing Transformers (Wu et al., 2022), and Lan et al. (2023) retrieve from *external* datastores, not the current context; recurrent memory layers (Bulatov et al., 2022) and Transformer-XL (Dai et al., 2019) extend context through segment-level caching. Both classes augment the backbone; our module is purely output-side and complementary, using only the current 1,024-token window.

Chang et al. (2023) revisit pointer architectures on pretrained GPT-2 and attribute their gains to relieving the softmax bottleneck (Yang et al., 2018; Chang & McCallum, 2022) rather than to retrieval, an explanation Section 6 probes with inference-time controls; relative to that work we extend the setting to five backbones, three without full attention, trained jointly from scratch. Boutkan et al. (2019) limit over-reliance on copying with pointer dropout, a role our gate warmup partly shares.

Fu et al. (2023), Jelassi et al. (2024), and Arora et al. (2024a) document recall deficits in fixed-state models; BASED (Arora et al., 2024b), Just Read Twice (Arora et al., 2024c), and HOLA (Cui, 2026) remedy them inside the backbone. Our module is an output-side analogue; we study its behavior rather than propose a new backbone, and claim no superiority. We use Mamba-1 (Gu & Dao, 2024) (not Mamba-2 (Dao & Gu, 2024)), a dense Jamba-style hybrid (not the MoE Jamba (Lieber et al., 2024)), Gated DeltaNet (Yang et al., 2025) (not GDN-2 (Hatamizadeh et al., 2026)), and only SISO Mamba-3 (Lahoti et al., 2026) (Appendix C).

3 Method

3.1 Copy Head Architecture

Throughout, t indexes *token position* within a sequence and u indexes the *training step*. Given a backbone that produces hidden states $\mathbf{h}_t \in \mathbb{R}^d$, the model at position t predicts the next token x_{t+1}; we write $w_t^\star = x_{t+1}$ for the target.

We project hidden states into a copy space of dimension d_{copy} ($= 256$ for most architectures; $= 768$ for GPT-2-style):

$$\mathbf{q}_t = W_q \mathbf{h}_t, \quad \mathbf{k}_t = W_k \mathbf{h}_t, \quad W_q, W_k \in \mathbb{R}^{d_{\text{copy}} \times d} \tag{1}$$

For each position t we compute attention scores over strictly preceding positions. Variant A (Mamba-1, Jamba-style, GDN) uses a sliding window of $W = 512$; Variant B (GPT-2-style, Llama-style) attends over the full causal prefix (up to $L = 1024$):

$$\alpha_{t,j} = \frac{\exp(\mathbf{q}_t^\top \mathbf{k}_j / \sqrt{d_{\text{copy}}})}{\sum_{j' \in \mathcal{W}(t)} \exp(\mathbf{q}_t^\top \mathbf{k}_{j'} / \sqrt{d_{\text{copy}}})}, \quad j \in \mathcal{W}(t) \tag{2}$$

where $\mathcal{W}(t) = \{\max(0, t-W), \ldots, t-1\}$ for Variant A and $\mathcal{W}(t) = \{0, \ldots, t-1\}$ for Variant B. The window excludes position t itself, so the current input token x_t—although valid causal context—is not a copy candidate; at $t = 0$ the window is empty and the copy distribution is masked to zero; for Variant A the implementation also sets $g_0 = 0$, so its mixture stays normalized (Variant B is treated below); a causality audit (six tests on GPT-2-style, one on each other family) finds no future-token access (Appendix L). The copy distribution aggregates probability mass onto source token identities:

$$p_{\text{copy}}(w \mid t) = \sum_{j \in \mathcal{W}(t):\, x_j = w} \alpha_{t,j} \tag{3}$$

We employ two mixing variants depending on the backbone.

Variant A (normalized, scalar sigmoid gate; Mamba-1, Jamba-style, GDN). A scalar gate logit $z_t = \mathbf{w}_g^\top \mathbf{h}_t + b_g$ gives $s_t = \sigma(z_t)$ and

$$p(w \mid t) = (1 - g_t) \cdot p_{\text{vocab}}(w \mid t) + g_t \cdot p_{\text{copy}}(w \mid t), \qquad g_t = c(u) \cdot s_t, \tag{4}$$

where $c(u) \in (0, c_{\max}]$ is a gate cap that ramps linearly with the training step u:

$$c(u) = \max\big(10^{-8},\, \min(c_{\max},\, u/N_{\text{ramp}})\big), \tag{5}$$

with $c_{\max} = 0.5$ and $N_{\text{ramp}} = 15{,}000$ micro-steps, so the cap rises from zero and reaches $c_{\max}$ at micro-step 7,500, 61M (GDN, micro-batch 8) to 123M tokens against ≈0.98B for the Variant B warmup (Table 4; the floor only keeps $\log c(0)$ finite); the mixture is exactly normalized, and in bfloat16 its $\log(1 - g_t)$ term must be computed as $\texttt{log1p}(-c\,\sigma(z_t))$ (Appendix B). The gate bias is initialized to $b_g = -3.0$ (gate ≈ 0.024 at the saturated cap). The ramp keeps the gate from spiking before the backbone's representations mature (not ablated; preliminary runs without it were unstable).

Variant B (learned softmax-gate mixture; GPT-2-style, Llama-style). A two-output gate network $f_g : \mathbb{R}^d \to \mathbb{R}^2$, implemented as $\text{Linear}(d, d/4) \to \text{GELU} \to \text{Linear}(d/4, 2)$ (148K parameters for $d = 768$), yields mixture weights $(\pi_{\text{vocab},t}, \pi_{\text{copy},t}) = \text{softmax}(f_g(\mathbf{h}_t))$:

$$p(w \mid t) = \pi_{\text{vocab},t} \cdot p_{\text{vocab}}(w \mid t) + \pi_{\text{copy},t} \cdot \tilde{p}_{\text{copy}}(w \mid t), \tag{6}$$

where $\tilde{p}_{\text{copy}}(w) = p_{\text{copy}}(w) + \epsilon$, $\epsilon = 10^{-10}$, keeps $\log \tilde{p}_{\text{copy}}$ finite for tokens absent from the context (so Variant B optimizes an approximately, not exactly, normalized likelihood; Appendix B). Mixing is done in log-space; the gate biases favor the vocabulary pathway at initialization ($b_{\text{vocab}} = 2.0$, $b_{\text{copy}} = -2.0$, so $\pi_{\text{copy}} = \sigma(-4) \approx 0.02$, with no upper bound thereafter); during warmup the copy logit is shifted by $\log\big(\max(10^{-10}, \min(1, u/N^B_{\text{warm}}))\big)$ with $N^B_{\text{warm}} = 15{,}000$ *optimizer* steps; at $t = 0$ the copy attention is zeroed but the gate is not, so the implementation keeps $\pi_{\text{copy},0} > 0$, the copy term contributes only the ϵ floor, and the mixture is subnormalized there ($Z_0 = 1 - \pi_{\text{copy},0} + \pi_{\text{copy},0}|V|\epsilon < 1$). LAMBADA never scores $t = 0$; lm-eval's rolling WikiText-103 windows score the first position of each window, where this deficit can only raise the Variant B models' perplexity, so it works against their reported gains; we have not quantified it.

Why two variants, and what the split costs. Variant B is an uncapped sigmoid gate (two-way softmax = sigmoid of the logit difference) with a two-layer network; Variant A is a capped linear sigmoid. Three differences are substantive: (i) *span*—Variant A is windowed to avoid introducing a quadratic term on linear-time backbones (truncating Variant B to W=512 retains 90–95% of the gain; Appendix Q); (ii) *cap*—$c_{\max}$=0.5 masks a bfloat16 failure (Appendix B); (iii) *warmup*—an 8–16× token difference from a units inconsistency, not tuning. The GPT-2-style retrain of Appendix R removes all three for one backbone. Gate form and warmup thus remain confounded with architecture class, so every claim is within a baseline/+copy pair (Sections 6, 7).

Cost. At B=1, L=1024 on one RTX 3090, the copy attention adds 1.0–2.5% to backbone prefill latency across the five 5B models (one timing pass each; the dense copy distribution our scoring backend materializes is a scoring artifact, not a deployment cost; Appendix P). Per generated token the head adds 0.3–1.5% of backbone FLOPs, and one cached decode step of the head costs 0.2–0.9% of the backbone step on two CPUs and 1.3–4.2% on a consumer RTX 3060, making the head a small compute addition for sub-billion models such as those deployed on edge devices, although it enlarges the Mamba-1 and GDN decode caches by 20–28% (Appendix S).

4 EXPERIMENTAL SETUP

We study five architecture families at 130M–198M parameters (a 1.5× range; Appendix C gives per-model counts). None is a standard published configuration, and we name them accordingly: "GPT-2-style" is a 12-layer, 768-wide pre-LN Transformer with ReLU MLPs, learned positions, and an *untied* output head (163M parameters); "Llama-style" is 12×768 with RoPE, SwiGLU, RMSNorm, and tied embeddings; "Mamba-1" uses 12 layers of width 1024; "Jamba-style" is a dense 12-layer hybrid with 10 Mamba-1 layers and 2 attention layers (positions 5 and 11); and "GDN" is Gated DeltaNet (Yang et al., 2025) as implemented in `flash-linear-attention`.

The copy head adds 394K parameters ($d_{\text{copy}} = 256$, sigmoid gate; Jamba-style and GDN), 0.5M (Mamba-1, because $d = 1024$; Llama-style, because of its 148K-parameter gate MLP), or 1.3M ($d_{\text{copy}} = d_{\text{model}} = 768$, a historical design choice, with the two-output gate MLP; GPT-2-style)—under 0.9% of any backbone.

All models are trained from scratch on a fixed 2B-token subset of FineWeb-Edu (Penedo et al., 2024) with GPT-2 BPE tokenization (50,257 vocab), sequence length 1024, effective batch size 64 sequences (65,536 tokens), a cosine learning-rate schedule (peak 6×10^{-4}), and AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$), at two token budgets, 2B and 5B, as separate runs with their own cosine schedules but a shared seed and data order; under the ≈20 tokens-per-parameter heuristic of Hoffmann et al. (2022), 2B is slightly under-trained and 5B is 1.25–1.9× compute-optimal (about 2.5 epochs over the subset); our baselines nonetheless remain far below published 180M-class models trained on 100B tokens (Table 14), so Section 5.6 tests a backbone with stronger native retrieval and Section 7 states what longer training leaves open. Weight decay, learning-rate floor, and micro-batching vary by architecture (Appendix A). We also train 350M Mamba-1 and GPT-2-style models (Appendix C) with 5B tokens. Variant B runs reserve a 2% holdout of the subset for training-time perplexity and Variant A runs sample from the full subset (both arms of every pair share this choice); all benchmarks are external to the subset.

We use lm-eval-harness (Gao et al., 2024) (0-shot) with eleven benchmarks: WikiText-103 (Merity et al., 2017) (word perplexity), LAMBADA (Paperno et al., 2016) (accuracy and perplexity), HellaSwag, PIQA, ARC-Easy, ARC-Challenge, WinoGrande, BoolQ, OpenBookQA, COPA, and SciQ. We further report perplexity on a true FineWeb-Edu holdout: 2M GPT-2 tokens (1,953 non-overlapping 1,024-token windows) streamed from the CC-MAIN-2024-51 crawl of FineWeb-Edu, added to the corpus after the sample-10BT release our subset was drawn from; it is a temporally separated holdout (we did not check for near-duplicates across crawls). Baselines are trained with the copy pathway disabled and scored through the same backend with its gate forced to zero. All 5B numbers come from one revision of this backend and one metric per task; an earlier revision that left the baselines' untrained gate parameters (present in the checkpoints, never trained; $g \approx 0.02$–0.05) active was corrected and every 5B baseline re-scored (Appendix P). Variant A models expose the gate cap c at inference (Variant B has none); every Variant A number in this paper uses the training cap $c = 0.5$ unless stated otherwise (exceptions, all at 2B or 350M, in Appendix P; sweep in Section 5.5).

5 RESULTS

5.1 THE COPY HEAD IMPROVES EVERY FAMILY ON WIKITEXT-103 AND LAMBADA

Table 1 reports the 5B-token results (Variant A at $c = 0.5$ throughout, Section 4); Figure 1 plots its LAMBADA and WikiText-103 columns next to the closed-gate scores of Section 5.4. In every pair the copy head lowers WikiText-103 perplexity and raises LAMBADA accuracy, as does the GPT-2-style backbone retrained with the SSM-family head (last row; Appendix R).[1] WikiText-103 perplexity falls by 14.6–21.7% (9.3–18.2% excluding the artifact), LAMBADA accuracy rises by 2.6–7.9 points, and LAMBADA perplexity falls by 31–76%. The same within-pair pattern holds at 2B tokens (Appendix D; not an independent replication, since the budgets share seed and data order). Of the remaining nine benchmarks, SciQ improves in all five pairs (+1.8 to +5.7 points)—the one

[1] Part of every WikiText-103 gain is a tokenization artifact: 1,178 `\n\n` tokens after section headings, a bigram absent from our training text, cost baselines 15–25 nats per token but copy models only 6.6–10.4 nats; the excl. column of Table 1 removes them, shrinking the reduction by an eighth to a half (Table 20).

downstream task whose support passage usually contains the answer, and the only non-headline task that collapses when the gate is closed (Section 5.4)—and BoolQ in four, while the other seven move within ±5.0 points with no consistent sign (Appendix F): outside retrieval-flavored tasks the head is neutral within single-seed noise.

5.2 General Next-Token Prediction on a True Holdout

WikiText-103 and LAMBADA both reward repeating tokens from context, so the last three columns of Table 1 score perplexity on the CC-MAIN-2024-51 holdout of Section 4 through the same backend. The copy head lowers it in every pair, by 9.9% and 8.2% for the two Transformers and 3.3–4.2% for the three SSM-family models: smaller than on WikiText-103, because FineWeb-Edu prose repeats far fewer tokens and the gain is confined to copyable tokens (Section 5.3), but of the same sign in every pair, so the head does not trade general perplexity for benchmark-specific retrieval.

5.3 Copy Gains Are Confined to Copyable Targets

Neither aggregate says whether the gain comes from retrieving token identities or from a more expressive output distribution. We therefore split every LAMBADA example by whether its target token occurs in the 512 tokens preceding it (the Variant A copy window) and score both halves through the backend of Table 1 (Appendix N). By this definition 73.7% of LAMBADA targets are copyable, and Table 18 shows a clean separation: on copyable examples the copy head raises accuracy by 4–11 points and lowers perplexity by 2–8× in every pair; on non-copyable examples accuracy is 1–4% with or without the head, and perplexity is *worse* with it in every pair—by 29–37% for the two Variant B models and 64–128% for the three Variant A models—because the gate assigns copy mass where nothing in the window matches. The same split on WikiText-103 (Table 19) agrees: per-token perplexity on copyable tokens (54% of the test set) falls 19–25%, and the reduction grows with distance to the source in every family, from 11–15% at 2–8 tokens to 28–36% at 129–512 tokens (Table 19), while on non-copyable tokens it moves by -5% (GPT-2-style) to $+6\%$ (Mamba-1). The aggregate gain is thus a retrieval gain minus a leakage cost, smaller for the Variant B gate (29–37% vs. 64–128% above).

5.4 Closing the Gate: The Output Pathway Offloads Retrieval to the Head

If the copy head were a redundant add-on, scoring the jointly trained model with its gate closed should return roughly baseline quality. It does not, in any pair, nor for the GPT-2-style backbone retrained with the Variant A head (last row). Table 2 (Figure 1, Appendix H) scores every 5B copy model with its copy pathway removed at inference—$c = 0$ for Variant A, the softmax gate bypassed for Variant B, leaving the backbone's vocabulary distribution—next to the independent baselines and the full model. With the gate closed, LAMBADA accuracy collapses to 3.6–6.5% (baselines 18.0–23.3%; the collapse replicates in all 14 three-seed 60M runs, Appendix W), LAMBADA perplexity rises 15–36× above baseline, and WikiText-103 perplexity lands 17–41% *above* it. For the three Variant A models, whose closed-gate WikiText scores we stratified, the loss is concentrated where the head operates: on copyable tokens the closed-gate backbones are 41–65% worse than their baselines, while on non-copyable tokens they are 6% *better*—and better than the full mixture, which leaks copy mass onto those tokens.

The other benchmarks barely move: with the gate closed, all five backbones stay within 1.6 points of their full models on the eight non-retrieval multiple-choice tasks (Appendix H); SciQ, whose support passage usually contains the answer, falls 8–15 points (closed 52.6–61.1% vs. full 67.2–71.0%). Joint training produces a division of labor: the vocabulary pathway stops modeling in-context repetition, which the head handles, and does slightly better on the remainder. A linear probe detects no loss in the representation: on WikiText-103, a linear probe on the final hidden state decodes copyable next tokens as accurately for the two +copy Transformer backbones as for their baselines (within one point), while their own LM heads lose 7–9 points on those tokens and gain 4–5 on the rest (Appendix V). For the three Variant A models, the closed-gate gain on non-copyable tokens is one the deployed mixture does not yet realize: with the gate open it is 3–6% *worse* than baseline on those tokens (Table 2), so the 6% is an oracle ceiling: recovering it would require a gate that knows whether the target occurs in the window, which is unavailable at inference. Offloading, and its confinement to copyable positions, are expected a priori (Boutkan et al. (2019)

Architecture	WikiText-103 PPL ↓				LAMBADA acc. (%) ↑			LAMBADA PPL ↓			FineWeb holdout PPL ↓		
	Base	+Copy	Δ%	excl. Δ%	Base	+Copy	Δpp	Base	+Copy	Δ%	Base	+Copy	Δ%
GPT-2-style TF	61.67	48.87	−20.8	−18.2	18.0	25.8	+7.8	366.4	87.7	−76.1	21.99	19.81	−9.9
Llama-style TF	55.68	43.59	−21.7	−16.8	21.3	29.2	+7.9	228.8	64.5	−71.8	19.70	18.09	−8.2
Mamba-1	52.94	43.70	−17.5	−9.3	23.3	25.9	+2.6	107.6	74.2	−31.1	18.32	17.72	−3.3
Jamba-style	49.35	40.99	−16.9	−10.6	22.3	27.5	+5.2	148.9	64.5	−56.7	17.83	17.24	−3.3
GDN	48.45	41.38	−14.6	−11.1	20.7	26.8	+6.1	172.9	69.5	−59.8	17.77	17.02	−4.2
GPT-2-style TF, Var. A head	61.67	52.25	−15.3	—	18.0	23.3	+5.3	366.4	100.9	−72.5	—	—	—

Table 1: **Main results at 5B tokens** (single seed). Each row is one controlled pair. Rows 1–2: Variant B; rows 3–5: Variant A at c=0.5; last row: GPT-2-style retrained with Variant A (Appendix R). The excl. column removes the \n\n artifact (Table 20). Magnitudes are not comparable across the A/B boundary.

report a pointer-generator over-relying on its pointer): under the mixture likelihood the vocabulary logits receive gradient scaled by $(1 - g_t)\, p_{\text{vocab}}(w_t^\star \mid t)/p(w_t^\star \mid t)$, the vocabulary pathway's share of responsibility for the target, which vanishes where the head is confident and equals one where nothing is copyable. What the mixture does not fix is the magnitude of the collapse or the sign on the remainder: the closed-gate backbone is not merely unharmed but better than its baseline on non-copyable tokens (single seed, three models). The finding is the size of the offloading and that residual gain, not its existence (a stop-gradient or auxiliary vocabulary-only loss, which would suppress it, was not run). We call it *output-pathway offloading* rather than lost capability because the deficit is measured at the output: the backbone's softmax no longer emits the copied identities, whether or not its hidden states still encode them (Appendix T).

5.5 Scaling to 350M and the Inference-Time Cap

At 350M with 5B tokens the head still lowers WikiText-103 perplexity: Mamba-1 by 19.2% (uncapped, as is its comparable 130M/5B figure of 11.2%; Figure 2) and GPT-2-style by 23.2% (20.8% at 130M), the relative gain growing from 130M/2B to 350M/5B as parameters and tokens grow together; LAMBADA rises $+2.5$ and $+6.5$pp (Appendix E; one seed per point; artifact included, not re-stratified at 350M).

Separately, sweeping the Variant A cap c at inference (Appendix G) shows WikiText-103 perplexity U-shaped in c with its minimum at the training cap and LAMBADA accuracy rising monotonically: raising c to 1.0 buys about 6 LAMBADA points for 5–8% higher WikiText-103 perplexity, a trade-off rather than an operating point, since caps were examined on the benchmarks.

5.6 One Copy Configuration, Two Backbones: Mamba-1 vs. Mamba-3 SISO

Comparisons across rows of Table 1 change the backbone family and, across the Variant A/B boundary, the head as well; this comparison holds the head fixed and changes only the SSM generation. We train Mamba-3 SISO (Lahoti et al., 2026), which addresses retrieval by backbone redesign (127M parameters; Appendix C), from scratch at 2B tokens with and without the *same* Variant A head as Mamba-1, on the same data, seed, and budget, scored through the backend of Table 1 at all five caps (Table 12, Appendix I).

Mamba-3 SISO's stronger native retrieval is visible at baseline (Table 12), and the same head still helps it on every headline metric: at the training cap, WikiText-103 perplexity falls 15.6% (55.3 → 46.7), LAMBADA accuracy rises 2.9 points, and LAMBADA perplexity falls 40% (126.6 → 76.2); uncapped, LAMBADA accuracy reaches 31.8% ($+8.3$ points) while WikiText-103 perplexity climbs back to 50.7. The causal source shuffle of Section 6 drops it to 3.5% LAMBADA and 75.3 WikiText-103, far below baseline: the gain is identity-specific here too. Scored uncapped like the 2B Mamba-1 pair (whose baseline used the earlier backend revision; Appendix P), its WikiText-103 reduction (-8.3%) matches Mamba-1's (-8.0%) while its LAMBADA gain ($+8.3$pp, a coincidence) is smaller than Mamba-1's $+12.1$pp: with the head held fixed, better native retrieval shrinks the deficit the head fills without making the head redundant (one seed, 2B tokens, no true-holdout or Mamba-2 control).

Architecture	LAMBADA acc. ↑			WT-103 word PPL ↓			WT copyable PPL ↓			WT non-copyable PPL ↓		
	Base	Closed	Full	Base	Closed	Full	Base	Closed	Full	Base	Closed	Full
GPT-2-style TF	18.0	**4.1**	25.8	61.7	**79.5**	48.9	9.38	—	7.03	185.5	—	176.5
Llama-style TF	21.3	**3.6**	29.2	55.7	**78.3**	43.6	8.71	—	6.59	159.7	—	155.6
Mamba-1	23.3	**5.0**	25.9	52.9	**66.3**	43.7	8.57	**14.18**	6.95	140.9	132.4	149.2
Jamba-style	22.3	**6.5**	27.5	49.4	**57.7**	41.0	8.08	**11.36**	6.54	135.9	128.3	140.3
GDN	20.7	**5.2**	26.8	48.4	**62.6**	41.4	8.46	**13.24**	6.71	132.7	124.6	138.9
GPT-2-style, Var. A head	18.0	**4.8**	23.3	61.7	**72.9**	52.3	9.38	—	—	185.5	—	—

Table 2: **Closing the copy gate at inference** (5B, single seed). Bold: closed-gate worse than baseline; underlined: better. Split not run for closed-gate Transformers or last row. Full eleven-task version in Appendix H.

6 Analysis

Section 5 established where the gain lands; this section asks what produces it, with two inference-time controls (identity corruption and a training-free cache), then turns to costs and to what can be said across pairs.

Table 3 re-scores every 5B copy model with the identity that each attended position j contributes to Eq. 3 replaced by that of a uniformly random position in $\{0, \ldots, j\}$ (never a later one; Appendix O). Attention and gate depend only on hidden states, never on the identities x_j, so they are untouched (the gate is near zero on most positions, median 0.02–0.11, and opens on a minority where it reaches the Variant A cap; Table 23) and the mixture receives the same copy mass on real context tokens—only the identities are wrong. The intervention tests dependence on the learned association between source positions and token identities, not retrieval against output expressivity (Chang et al., 2023), since a context-conditioned output distribution can also depend on correct identities. The dependence is total: LAMBADA accuracy falls to 2–4% against baselines of 18–23%, LAMBADA perplexity rises 10–44× above baseline, and WikiText-103 perplexity lands 21–67% above it. From worst to best, the order is shuffled, closed, baseline, full on both LAMBADA accuracy and WikiText-103 perplexity for all five models (Mamba-1 WikiText-103: 72.1, 66.3, 52.9, 43.7; Table 2); the one exception is LAMBADA perplexity for the three Variant A models, where shuffled is less bad than closed (Appendix O). Wrong identities thus do active harm beyond exposing the offloaded backbone. None of the head's gain survives wrong identities at inference; whether expressivity also contributed during training is not separable this way (Section 7).

A training-free neural cache (Grave et al., 2017) on the two Transformer 5B baselines (cosine attention over hidden states, each paired with the token that followed it; mixing weight and temperature swept; 512-token window) reduces WikiText-103 word perplexity by up to 20.3% (GPT-2-style) and 17.1% (Llama-style) and raises LAMBADA accuracy by at most 0.8 and 1.3 points. This standalone scorer replicates neither lm-eval's multi-token-word LAMBADA protocol nor its WikiText detokenization (GPT-2-style baseline: 94.7 perplexity, 35.0% LAMBADA), so we compare only within-scorer changes. Under that same scorer, the trained copy models reduce WikiText-103 perplexity by 40.8% (GPT-2) and 38.2% (Llama) and raise LAMBADA accuracy by 6.2 and 6.1 points: the cache recovers about half of the perplexity gain and 13–21% of the accuracy gain (Appendix U). A variant that pairs each state with its own input token, as our head does, recovers less (11.8% and 8.5%; at most $+0.1$ points).

On the synthetic suite of Appendix N (2B results: Appendix M) (Table 21), the 5B heads raise state-tracking accuracy (a cloze probe of Appendix N) by 10–25 points in four pairs but lower MQAR-update—multi-query associative recall (Arora et al., 2024a) of a value after an in-context correction, *i.e.* retraction of the superseded value—in three (GDN 82.5→28.0%, with the superseded value winning 64% of items once the head is added), while the GPT-2-style pair, whose baseline is among the weakest retrievers, instead gains 22 points (about 4.5 standard errors). A head that retrieves whichever value co-occurred with a key supplies retrieval where the backbone lacks it and competes with retraction where it already has it (Appendix N).

The true-holdout reduction is ordered Transformers (-8–10%) > SSM-family (-3–4%), but because the head changes with architecture class (Section 3), none of this is attributable to the backbone. Truncating the Variant B window to $W = 512$ retains 90–95% of the gain (Appendix Q), so

Architecture	LAMBADA acc. (%) ↑			LAMBADA PPL ↓			WikiText-103 PPL ↓		
	Base	+Copy	Shuffled	Base	+Copy	Shuffled	Base	+Copy	Shuffled
GPT-2-style TF	18.0	25.8	**2.4**	366	87.7	**7587**	61.7	48.9	**94.9**
Llama-style TF	21.3	29.2	**2.2**	229	64.5	**10139**	55.7	43.6	**93.0**
Mamba-1	23.3	26.0	**3.3**	108	74.2	**1745**	52.9	43.7	**72.1**
Jamba-style	22.3	27.5	**3.9**	149	64.5	**1517**	49.4	41.0	**59.7**
GDN	20.7	26.8	**3.4**	173	69.5	**2132**	48.4	41.4	**66.2**
GPT-2-style, Var. A head	18.0	23.3	**2.9**	366	100.9	—	61.7	52.3	**76.8**

Table 3: **Causal source-shuffle control** (5B, single seed). Each attended position contributes a random earlier token's identity; attention and gate unchanged. Bold: every shuffled model falls below its baseline.

span accounts for at most 5–10%. Retraining GPT-2-style with the Variant A head (Appendix R) retains 68–74% of the Variant B gain and places it inside the SSM-family range at the same cap: the Transformers-first WikiText-103 and LAMBADA ordering does not survive head equalization, and whether the holdout ordering does is untested (one backbone, one seed). Among the three Variant A pairs, the Jamba-style hybrid, which contains two full attention layers, has no baseline LAMBADA advantage over Mamba-1 (22.3 vs. 23.3%) and gains at least as much from the head ($+5.2$ vs. $+2.6$pp): at this scale two in-backbone attention layers do not make an output-side copy path redundant, though we have no matched single-attention-layer control.

7 Limitations

Gate form, warmup, and d_{copy} differ between Transformers and SSM-family models; span accounts for 5–10%, and the GPT-2 Variant A retrain (Appendix R) retains 68–74% of Variant B's gains on both headline metrics. We rank no architecture; within-pair conclusions stand.

All 130M–350M results use one seed (42). At 60M, three seeds reproduce every headline gain for all five families (all four metrics in all 15 seed pairs, which are correlated; Appendix K); neither the 60M nor the 45M two-seed study (Appendix J) is at 130M scale.

The 45M capacity controls use a Variant B Transformer only; at 130M the identity control is inference-time only; the neural cache (Section 6) covers the two Transformer baselines only; the closed-gate copyability split covers only SSM-family models.

Appendix P collects all recording asymmetries. We study joint training from scratch at 130M–350M only, the regime of edge-deployed models; joint training at 1B+ is untested, and on-device latency and quantized inference are unmeasured. In preliminary frozen-backbone retrofits whose raw outputs were not archived, the head gained at most 0.9 LAMBADA points on Mamba-1.4B, Llama-3.2-1B, and Qwen2.5-1.5B, within 0.2 points of a neural cache, and on Qwen2.5-7B-Instruct its gate stayed near zero with unchanged perplexity, which suggests, for those settings only, that the head adds little without joint training.

8 Conclusion

Across five families and four scales, jointly training a compact copy head reveals output-pathway offloading and identity-specific gains. In the two Transformers we probed, our linear probe detects no reduction in the decodability of copy targets from the backbone's hidden state, while its native readout becomes less accurate on them; a jointly trained copy head is part of the model rather than an optional add-on: removing it costs more than never having it, and ablating such a module measures how training reallocated a skill, not what the module adds.

Ethics Statement

The copy head retrieves tokens from the *current* input context, which could amplify verbatim reproduction of sensitive, private, or copyrighted text present in the prompt (for example, in retrieval-augmented or multi-user settings). This is distinct from memorization of training data, which the

copy head neither directly enables nor prevents; we have not measured its effect on training-data extraction. We recommend standard guardrails when deploying models with copy mechanisms in applications involving private data.

Reproducibility Statement

The supplementary material contains all training scripts, model configurations, evaluation code, and result files, and lists the few raw outputs that were not archived. Training data is a fixed 2B-token subset of FineWeb-Edu (Penedo et al., 2024), publicly available; the download and tokenization scripts and the CC-MAIN-2024-51 holdout construction are included. Evaluation uses lm-eval-harness (Gao et al., 2024); the copyability-stratified scorer, the causal source-shuffle and gate-statistics options of the evaluation backend, and the synthetic probes of Appendix N are included and reproduce the lm-eval totals within 0.3% (one case 1.4%; Appendix P). A six-test causality audit verifying the copy head introduces no future-token leakage is in Appendix L. Hyperparameters are in Appendix A; all numbers behind the figures are tabulated in Appendices D–K, M–O, and Q–R.

Declaration on Generative AI

ChatGPT and Claude Opus have been used to enhance grammar, rectify spelling mistakes, and rephrase writing for clarity. All AI-generated suggestions were reviewed and verified by the authors. The authors take full responsibility for the content of this work.

References

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations (ICLR)*, 2024a.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In *International Conference on Machine Learning (ICML)*, 2024b.

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*, 2024c.

David Bamman, Sejal Popat, and Sheng Shen. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2138–2144, 2019.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning (ICML)*, 2022.

Freek Boutkan, Jorn Ranzijn, David Rau, and Eelco van der Wel. Point-less: More abstractive summarization with pointer-generator networks. *arXiv preprint arXiv:1905.01975*, 2019.

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2207.06881.

Haw-Shiuan Chang and Andrew McCallum. Softmax bottleneck makes language models unable to represent multi-mode word distributions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Haw-Shiuan Chang, Zonghai Yao, Alolika Gon, Hong Yu, and Andrew McCallum. Revisiting the architectures like pointer networks to efficiently improve the next word distribution, summarization factuality, and beyond. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12707–12730, 2023.

Wanyun Cui. A hippocampus for linear attention: An exact memory for what the recurrent state forgets. *arXiv preprint arXiv:2607.02303*, 2026.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

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.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry Hungry Hippos: Towards language modeling with state space models. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2212.14052.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, et al. The language model evaluation harness, 2024. Zenodo, doi:10.5281/zenodo.12608602.

Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1612.04426.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling (COLM)*, 2024. arXiv:2312.00752.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 30016–30030, 2022.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14, 2014. doi: 10.1109/ISSCC.2014.6757323.

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 *First Conference on Language Modeling (COLM)*, 2024. arXiv:2404.06654.

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.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations (ICLR)*, 2020.

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. arXiv:2603.15569.

Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. Copy is all you need. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2307.06962.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid Transformer-Mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1609.07843.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *Transformer Circuits Thread; arXiv:2209.11895*, 2022.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

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 *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Erik Wijmans, Brody Huval, Alexander Hertzberg, Vladlen Koltun, and Philipp Krähenbühl. Cut your losses in large-vocabulary language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2411.09009.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2412.06464.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations (ICLR)*, 2018.

A Hyperparameters

Shared hyperparameters. All architectures: sequence length 1024, effective batch size 64 sequences (65,536 tokens), peak LR 6×10^{-4} (cosine decay over the run), AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$), gradient clipping 1.0, GPT-2 BPE tokenizer (50,257 vocab), bfloat16 autocast with gate/mixing in float32, seed 42. No auxiliary copy loss is applied in any 130M experiment. Learning-rate schedules and learning-rate warmups are indexed by micro-step in every script; the gate warmup is indexed by optimizer step in the Variant B scripts and by micro-step in the Variant A scripts (Table 4).

Parameter	GPT-2-style	Llama-style	Mamba-1	Jamba-style	GDN	Mamba-3 SISO
Weight decay	0.1	0.1	0.01	0.1	0.01	0.01
LR warmup (micro-steps)	2000	2000	500	500	500	500
LR floor	5×10^{-5}	5×10^{-5}	1×10^{-5}	1×10^{-5}	6×10^{-5}	1×10^{-5}
Micro-batch × grad accum (5B)	4×16 / 16×4 †	16×4	16×4	16×4	8×8	(2B only)
Data holdout	2%	2%	none	none	none	none
Copy head configuration:						
Copy variant	B	B	A	A	A	A
d_{copy}	768	256	256	256	256	256
Copy window	full*	full*	512	512	512	512
Gate type	softmax MLP	softmax MLP	sigmoid	sigmoid	sigmoid	sigmoid
Gate cap $c_{\max}$	—	—	0.5	0.5	0.5	0.5
Gate warmup horizon	15K^{opt}	15K^{opt}	7.5K^{μ}	7.5K^{μ}	7.5K^{μ}	7.5K^{μ}
Gate warmup in tokens	≈0.98B	≈0.98B	≈123M	≈123M	≈61M	≈123M§
Gate init bias	$b_{\text{copy}}=-2$	$b_{\text{copy}}=-2$	$b_g=-3$	$b_g=-3$	$b_g=-3$	$b_g=-3$

Table 4: Per-architecture training and copy-head hyperparameters. ^{opt} = optimizer steps (15K optimizer steps × 65,536 tokens); $^{\mu}$ = micro-steps (the Variant A ramp constant is $N_{\text{ramp}} = 15\text{K}$ micro-steps, which with $c_{\max} = 0.5$ saturates at 7.5K micro-steps, Eq. 5; token counts use the 5B-run micro-batch sizes). * Full causal prefix (up to $L = 1024$), not windowed. § Assuming micro-batch 16. The 2B runs used the same effective batch with micro-batch sizes chosen per GPU. † The GPT-2-style 5B +copy run used micro-batch 4 × accumulation 16 and its baseline 16 × 4 (same 64-sequence effective batch, 5B tokens, seed, data order, and 76,293 optimizer steps), so the shared 2000-micro-step LR warmup spans 125 optimizer steps for the +copy run and 500 for the baseline—the one training asymmetry in that pair; Llama-style (accumulation 4) warms up over 500 optimizer steps for both runs.

B Numerical Stability and Loss Computation

The logsigmoid formulation. In bfloat16, when the gate logit $z_t \geq 6.3$, $\sigma(z_t)$ rounds to exactly 1.0; the naive $\log(1-g_t)$ then evaluates $\log 0 = -\infty$, whose backward pass yields NaN gradients that gradient clipping propagates to all parameters. We observed this failure in the Mamba-1 configuration ($d = 1024$), whose gate logits saturated earlier than those of narrower models. The fix depends on the cap. When $c \geq 1.0$ we replace $\log(1-\sigma(z))$ with the numerically equivalent $\texttt{logsigmoid}(-z)$, computed in float32 with autocast disabled. When $c < 1.0$ (*e.g.* $c_{\max} = 0.5$) we compute $\log(1 - c\cdot\sigma(z))$ via $\texttt{log1p}(-c\cdot\sigma(z))$ in float32, which is safe because $1 - g_t \geq 1 - c_{\max} > 0$. We did not ablate whether the $c_{\max} = 0.5$ runs would have failed without it.

Variant B normalization. The floored copy distribution of Variant B sums to $1 + |V|\epsilon \approx 1 + 5\times 10^{-6}$, so its likelihood is approximately rather than exactly normalized; the Llama implementation floors only the target's copy probability, with the same effect.

Target-only loss computation. The mixed log-probability $\log p(w_t^\star \mid t)$ is needed only at the ground-truth target. The copy probability $p_{\text{copy}}(w_t^\star \mid t)$ is gathered from copy scores at positions whose token matches the target, and mixing is performed in log-space at the target only. For the vocabulary distribution, the Jamba-style and Mamba-3 implementations use a chunked fused GEMM + cross-entropy (chunk size 4096 tokens; cf. Cut Your Losses, Wijmans et al. 2025) that avoids materializing the full (B, L, V) logit tensor, reducing peak memory from $\mathcal{O}(BLV)$ to $\mathcal{O}(\text{chunk}\cdot V + BLW)$; the GPT-2, Llama, Mamba-1, and GDN implementations materialize the full logits. The chunked approach is an engineering convenience used where memory was binding, not a general contribution of the cross-architecture system.

C Per-Architecture Configuration

Table 5 lists the per-architecture configurations and copy-head variants.

Architecture	Params	+Copy	d_{model}	Layers	Heads	MLP	State	Var.	Notes
130M class									
GPT-2-style	163.0M	+1.3M	768	12	12	3072	—	B	pre-LN, ReLU, abs. pos., untied
Llama-style	180.6M	+0.5M	768	12	12	4096	—	B	SwiGLU, RoPE, RMSNorm, tied
Mamba-1	131.5M	+0.5M	1024	12	—	—	16	A	d_{inner}=2048, conv 4, tied
Jamba-style	177.1M	+0.4M	768	12	12*	3072	16	A	10 Mamba-1 + 2 attn*, GELU, untied
GDN	198.3M	+0.4M	768	12	12	3072	h_d=64	A	head dim 64, expand$_v$ 1, conv 4
Mamba-3 SISO	127.0M	+0.4M	768	12	24	1536	128	A	gated MLP, head dim 64, exp. 2, tied
350M class									
GPT-2-style	354.8M	+2.4M	1024	24	16	4096	—	B	GELU MLP, d_{copy}=1024
Mamba-1	313.5M	+0.7M	1280	24	—	—	16	A	mamba-ssm defaults

Table 5: Per-architecture configurations. "+Copy" is the copy-head parameter overhead: 394K for $d_{\text{copy}} = 256$ with the sigmoid gate (Variant A), 0.5M for Mamba-1 ($d_{\text{model}} = 1024$) and Llama-style ($d_{\text{copy}} = 256$ with the two-output gate MLP), and 1.3M for GPT-2-style ($d_{\text{copy}} = d_{\text{model}} = 768$, a historical design choice). "Var." is the gate variant of Section 3: A = capped sigmoid gate, B = softmax-gate mixture. *Jamba-style attention layers are at positions 5 and 11; the head count applies to those two layers only. MLP column shows the inner dimension. All architectures use the GPT-2 BPE tokenizer. Initialization: PyTorch defaults except Mamba-1 and Mamba-3, which use the official `mamba-ssm` initialization. The 130M-class models span 127M–198M parameters because embedding tying, MLP width, and layer type differ; none is a standard published recipe (Section 4). GDN is Gated DeltaNet (Yang et al., 2025), not Gated DeltaNet-2.

D 2B-Token Results

Table 6 presents the 2B-token evaluation for all five architectures. The 2B runs predate the cap sweep and were scored uncapped ($c = 1.0$) only, an exception to the $c = 0.5$ protocol of Section 4; they were not evaluated on the true holdout of Table 1.

	WT-103 PPL ↓		LAMBADA ↑		HellaSwag ↑		BoolQ ↑		SciQ ↑	
Arch	B	+C	B	+C	B	+C	B	+C	B	+C
GPT-2-style	69.7	56.9	15.9	22.2	28.1	29.2	59.2	61.8	72.1	79.4
Llama-style	96.7	56.8	16.6	23.4	27.7	30.1	39.1	61.2	58.8	75.1
Mamba-1	61.6	56.7	18.2	30.3	30.7	30.9	62.1	61.2	74.5	80.3
Jamba-style	96.4	56.5	15.1	29.6	29.5	30.2	49.4	62.2	71.8	78.4
GDN	61.3	49.0	17.2	25.1	30.7	31.6	61.1	57.7	77.3	77.4

Table 6: 2B-token evaluation (130M-class, single seed; five of the eleven benchmarks shown). All five pairs improve on WikiText-103 and LAMBADA (LAMBADA: +6.3pp GPT-2-style, +6.8pp Llama-style, +12.1pp Mamba-1, +14.5pp Jamba-style, +7.9pp GDN); magnitudes are not comparable across the Variant A/B boundary (Section 3). The Llama-style and Jamba-style 2B baselines converged poorly (WikiText PPL above 96), which inflates their 2B gains. The Variant A baselines in this table were evaluated with the earlier backend revision of Section 4, which at 5B overstated the Jamba-style and GDN baseline WikiText-103 perplexities by 12% and 9% (Appendix P), so their 2B gains are likely overstated too. Because the two budgets share seed and data order, agreement with the 5B results is consistency across training durations, not independent replication.

E 350M-Scale Results

Table 7 tabulates the 350M-scale runs of Section 5.5; Figure 2 plots their WikiText-103 change against the 130M runs. At 350M with 5B tokens the copy head reduces WikiText-103 perplexity for Mamba-1 ($47.73 \to 38.57$, -19.2%) and GPT-2-style ($60.3 \to 46.3$, -23.2%), the relative gain growing monotonically from 130M/2B to 350M/5B for both (Mamba-1 along its uncapped series; the 350M Mamba-1 pair was scored only at $c = 1.0$, an exception to the protocol of Section 4); LAMBADA rises +2.5pp and +6.5pp, and with one seed per configuration we cannot separate scale effects from run-to-run variance.

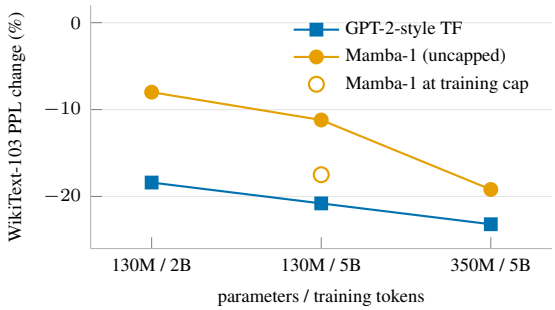

Figure 2: Relative WikiText-103 improvement from the copy head grows with parameters and tokens for both families that we scaled (single seed per point; Table 7 and Appendix D). The Mamba-1 line is its uncapped series, the only one measured at all three points; the hollow marker is the 130M/5B point at the training cap $c = 0.5$ (Table 1).

Model	WT-103 PPL	LAMBADA	HellaSwag	ARC-E	SciQ
Mamba-1 350M	47.73	25.7%	31.9%	53.3%	78.1%
Mamba-1 350M+Copy	**38.57**	**28.2%**	**37.5%**	**54.7%**	**81.1%**
Δ	−19.2%	+2.5pp	+5.6pp	+1.4pp	+3.0pp
GPT-2-style 350M	60.3	20.0%	31.1%	47.4%	74.4%
GPT-2-style 350M+Copy	**46.3**	**26.5%**	**32.2%**	47.4%	**77.9%**
Δ	−23.2%	+6.5pp	+1.1pp	+0.0pp	+3.5pp

Table 7: 350M-scale results (5B tokens, single seed). GPT-2-style uses the Variant B head; the Mamba-1 pair uses Variant A and was scored uncapped ($c = 1.0$) only, so its LAMBADA gain is not comparable with the $c = 0.5$ rows of Table 1: at 130M Mamba-1's uncapped LAMBADA gain is +8.8pp against +2.6pp at the training cap (Appendix G), and its HellaSwag gain +0.5pp (Appendix F).

F Full Eleven-Benchmark Results at 5B Tokens

Table 8 gives all eleven benchmarks at 5B tokens; its two headline rows are those of Table 1. The GPT-2-style backbone retrained with the Variant A head is in Appendix R.

Benchmark	GPT-2-style B	GPT-2-style +C	Llama-style B	Llama-style +C	Mamba-1 B	Mamba-1 +C	Jamba-style B	Jamba-style +C	GDN B	GDN +C
WT-103 PPL↓	61.7	48.9	55.7	43.6	52.9	43.7	49.4	41.0	48.4	41.4
LAMBADA acc.↑	18.0	25.8	21.3	29.2	23.3	25.9	22.3	27.5	20.7	26.8
LAMBADA PPL↓	366	87.7	229	64.5	108	74.2	149	64.5	173	69.5
HellaSwag† ↑	29.6	30.7	31.3	33.0	32.9	33.4	32.8	33.3	33.3	34.7
PIQA† ↑	60.6	62.0	62.1	62.6	64.4	63.7	61.6	63.4	63.1	62.9
ARC-E† ↑	44.8	46.2	45.8	45.4	46.1	47.3	45.6	44.9	45.5	45.4
ARC-C† ↑	23.0	25.0	25.3	25.0	24.0	25.7	25.1	25.0	25.6	26.5
WinoGrande↑	50.4	51.6	52.0	50.8	48.9	52.2	50.3	49.8	51.3	50.3
BoolQ↑	54.9	61.2	48.4	60.7	60.2	59.1	58.4	61.6	58.2	62.1
OBQA† ↑	31.6	31.2	30.4	28.6	30.8	31.8	31.4	33.4	32.4	32.4
COPA↑	65.0	63.0	63.0	63.0	68.0	63.0	66.0	63.0	64.0	63.0
SciQ† ↑	65.3	71.0	64.6	67.2	66.7	70.1	66.7	68.5	65.2	69.5

Table 8: Complete 11-benchmark evaluation at 5B tokens (single seed; one backend, lm-eval-harness 0-shot, baselines scored with the gate forced to zero, Variant A copy models at the training cap $c = 0.5$). B = baseline, +C = with copy head. Every baseline/+copy pair uses the same metric within each row: † = `acc_norm` (length-normalized; HellaSwag, PIQA, ARC, OpenBookQA, SciQ), unmarked accuracy rows report `acc`. WikiText-103, LAMBADA, and SciQ improve in every pair and BoolQ in four; the other seven tasks move within ±5.0 points with no consistent sign (COPA has only 100 examples, so its changes are 0–5 items). No confidence intervals or multiple-comparison corrections are applied.

G CAP-SWEEP TABLES

The sweep of Section 5.5 varies the Variant A gate cap c at inference with weights fixed; Figure 3 plots it and Tables 9 and 10 tabulate its values. WikiText-103 perplexity is U-shaped in c for all three Variant A models, with its minimum at the training cap, whereas LAMBADA accuracy rises monotonically with c (Jamba-style and GDN at five caps; Mamba-1 at $c = 0.5$ and $c = 1.0$): raising the cap above its training value buys LAMBADA accuracy with perplexity, so the pathway is load-bearing and the best operating point is task-dependent. The $c = 0.5$ column reproduces Table 1; the other caps come from the same sweep pipeline (an earlier backend revision gave 48.00 for Mamba-1 and 43.49 for Jamba-style at $c = 1.0$). Against the clean baselines of Table 1, the WikiText-103 improvements are -17.5% (Mamba-1), -16.9% (Jamba-style), and -14.6% (GDN) at the training cap versus -11.2%, -12.5%, and -8.0% uncapped, and the LAMBADA gains are $+2.6$, $+5.2$, and $+6.1$pp at the training cap versus $+8.8$, $+11.5$, and $+12.3$pp uncapped. Because the caps were examined on the benchmarks themselves, we report the uncapped point as a trade-off and use $c = 0.5$ throughout. The 2B Mamba-3 SISO copy model, swept over the same five caps (Table 13), shows the same profile: WikiText-103 is minimized at the training cap (46.67, against 50.69 uncapped) and LAMBADA accuracy rises monotonically from 10.4% at $c = 0.1$ to 31.8% at $c = 1.0$.

Cap c	0.10	0.25	0.50	0.75	1.00	Pattern
Variant A (normalized, cap applies):						
Mamba-1	48.5	45.4	**43.7**	43.8	47.0	U-shaped, 0.50 optimal
Jamba-style	44.3	42.1	**41.0**	41.1	43.2	U-shaped, 0.50 optimal
GDN	45.5	42.8	**41.4**	41.5	44.6	U-shaped, 0.50 optimal
Variant B (softmax-gate mixture, no cap parameter):						
GPT-2-style	48.9	48.9	48.9	48.9	48.9	Invariant by construction
Llama-style	43.6	43.6	43.6	43.6	43.6	Invariant by construction

Table 9: WikiText-103 perplexity under varying inference-time gate cap (5B models). Variant B's invariance is a property of the formulation and carries no information about gate magnitude.

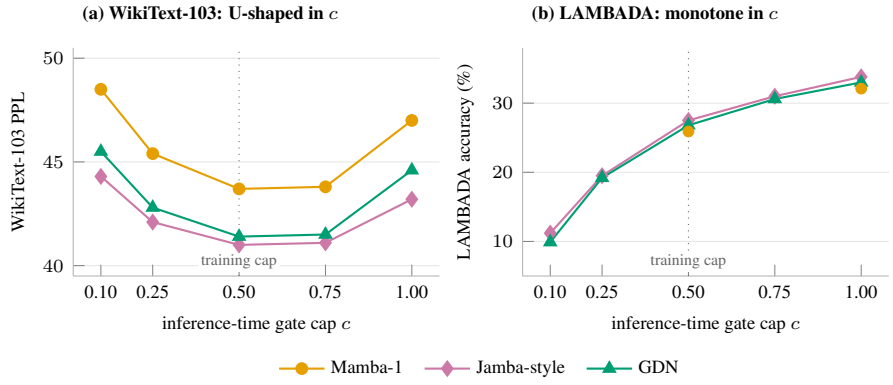

Figure 3: **Inference-time gate-cap sweep** on the 5B models (weights fixed, c varied). **(a)** WikiText-103 perplexity for the three Variant A models is U-shaped with its minimum at the training cap; the two Variant B Transformers are omitted because their softmax-gate mixture has no cap parameter and their perplexity is invariant to c by construction (Table 9). **(b)** LAMBADA accuracy for the Jamba-style and GDN models rises monotonically with c; the two Mamba-1 points ($c = 0.5$ and 1.0) are its only LAMBADA measurements.

Cap c	0.10	0.25	0.50	0.75	1.00
Jamba-style LAMBADA	11.2%	19.5%	27.5%	31.0%	**33.8%**
GDN LAMBADA	9.9%	19.2%	26.8%	30.6%	**33.0%**
Mamba-1 LAMBADA	—	—	25.9%	—	32.1%

Table 10: LAMBADA accuracy vs. cap (5B models). Mamba-1 was swept on WikiText only; its LAMBADA accuracy was measured at $c = 1.0$ (sweep pipeline) and at $c = 0.5$ (the 11-task pass of Table 1, same backend), not at the other caps. The $c = 0.5$ column equals Table 1. The cap values reported here were examined on the benchmarks themselves; a deployment would need to select c on held-out validation data.

H Closed-Gate Evaluation on All Eleven Benchmarks

Figure 1 (Section 1) plots the LAMBADA and WikiText-103 columns of Table 2. Table 11 scores all five 5B copy models on the full benchmark suite with the copy pathway removed at inference (Variant A at gate cap zero; the Transformers with their softmax gate bypassed, so $p = p_{\text{vocab}}$), next to the independently trained baselines and the full models of Table 8, under the same backend and metric convention. For all five, the eight non-retrieval multiple-choice benchmarks move by at most 1.6 points between the closed-gate and full models (the largest move is Llama-style HellaSwag, 31.4 vs. 33.0; the median over the 40 cells is 0.3), while LAMBADA, WikiText-103, and SciQ—the three tasks whose targets are usually present in the context—collapse.

Model	Scoring	WT-103↓	LAMB. acc↑	LAMB. PPL↓	SciQ†	HS†	PIQA†	ARC-E†	ARC-C†	Wino	BoolQ	OBQA†	COPA
GPT-2-style TF	Base	61.7	18.0	366	65.3	29.6	60.6	44.8	23.0	50.4	54.9	31.6	65.0
	Closed	**79.5**	**4.1**	**7096**	**60.4**	29.7	61.5	46.3	25.9	51.5	61.2	31.6	63.0
	Full	48.9	25.8	87.7	71.0	30.7	62.0	46.2	25.0	51.6	61.2	31.2	63.0
Llama-style TF	Base	55.7	21.3	229	64.6	31.3	62.1	45.8	25.3	52.0	48.4	30.4	63.0
	Closed	**78.3**	**3.6**	**8341**	**52.6**	31.4	62.4	45.2	25.9	51.4	60.6	29.2	63.0
	Full	43.6	29.2	64.5	67.2	33.0	62.6	45.4	25.0	50.8	60.7	28.6	63.0
Mamba-1	Base	52.9	23.3	108	66.7	32.9	64.4	46.1	24.0	48.9	60.2	30.8	68.0
	Closed	**66.3**	**5.0**	**2431**	**61.1**	32.4	63.7	47.4	26.0	52.4	59.2	32.0	62.0
	Full	43.7	25.9	74.2	70.1	33.4	63.7	47.3	25.7	52.2	59.1	31.8	63.0
Jamba-style	Base	49.4	22.3	149	66.7	32.8	61.6	45.6	25.1	50.3	58.4	31.4	66.0
	Closed	**57.7**	**6.5**	**2190**	**60.3**	32.2	62.6	45.0	25.3	49.3	61.7	34.0	64.0
	Full	41.0	27.5	64.5	68.5	33.3	63.4	44.9	25.0	49.8	61.6	33.4	63.0
GDN	Base	48.4	20.7	173	65.2	33.3	63.1	45.5	25.6	51.3	58.2	32.4	64.0
	Closed	**62.6**	**5.2**	**3203**	**60.6**	33.5	62.7	45.7	27.1	51.5	62.2	32.6	63.0
	Full	41.4	26.8	69.5	69.5	34.7	62.9	45.4	26.5	50.3	62.1	32.4	63.0

Table 11: **Closed-gate evaluation on all eleven benchmarks for all five pairs** (5B models, single seed; lm-eval-harness 0-shot, one backend). Base and Full rows are those of Table 8 (Variant A at $c = 0.5$); Closed is the same +copy checkpoint with its copy pathway removed at inference. † = `acc_norm`; other accuracies are `acc`; every column uses one metric. Bold marks the tasks on which the closed-gate backbone falls well below the independent baseline; on SciQ the support passage usually contains the answer. On the eight remaining tasks the closed-gate and full models differ by at most 1.6 points. COPA has 100 examples, so Mamba-1's -6-point gap to its baseline is six items.

I Mamba-3 SISO: Controlled Comparison and Cross-Paper Context

Table 12 is the controlled Mamba-1 vs. Mamba-3 SISO comparison discussed in Section 5.6, Table 13 sweeps the inference-time cap of the Mamba-3 SISO copy model, and Table 14 places our Mamba-1 numbers next to the published Mamba-2 and Mamba-3 results, which use $20\times$ more tokens and a different tokenizer and context length.

Model (2B tokens)	LAMBADA acc↑	LAMBADA PPL↓	WT-103 PPL↓
Mamba-1 baseline (131M)	18.2%	—	61.6
Mamba-1 + Copy (132M), uncapped ($c = 1.0$)	**30.3%**	—	**56.7**
Mamba-3 SISO baseline (127M)	23.5%	126.6	55.3
Mamba-3 SISO + Copy (127M), training cap ($c = 0.5$)	26.4%	76.2	**46.7**
Mamba-3 SISO + Copy (127M), uncapped ($c = 1.0$)	**31.8%**	**50.1**	50.7
Mamba-3 SISO + Copy (127M), $c = 0.5$, causal source shuffle	3.5%	1731	75.3

Table 12: **Within-family controlled comparison at 2B tokens** (single seed; parameter counts in parentheses). Both pairs use the identical Variant A copy head, data, data order, seed, and token budget, and all rows are scored through the backend of Table 1. The head helps both backbones on every headline metric; with the retrieved identities randomized (Section 6) Mamba-3 SISO falls far below its baseline. Evaluation asymmetries: the Mamba-1 pair was scored uncapped and its baseline with the earlier backend revision (Appendix P); Mamba-3 SISO's vocabulary distribution is renormalized over the 50,257 real token ids (training used a 50,272-padded vocabulary). The 2B runs were not evaluated on the true FineWeb-Edu holdout of Table 1.

Cap c	0.10	0.25	0.50	0.75	1.00
WikiText-103 PPL↓	51.39	48.26	**46.67**	46.91	50.69
LAMBADA acc.↑	10.4%	18.7%	26.4%	29.7%	**31.8%**
LAMBADA PPL↓	205.7	118.4	76.2	58.8	**50.1**

Table 13: **Inference-time cap sweep of the 2B Mamba-3 SISO copy model** (weights fixed; baseline 23.5% / 126.6 / 55.30). The profile matches the 5B Variant A models (Appendix G): WikiText-103 is minimized at the training cap and LAMBADA accuracy rises monotonically with c. At $c = 0.5$ the mean effective gate on WikiText-103 is 0.154, against 0.160 for the 5B Mamba-1 model (Table 23). Cap values were examined on the benchmarks themselves.

Model	Params	Tokens	LAMBADA PPL ↓	Protocol
Mamba-1 baseline	131M	5B	107.6	Ours (GPT-2 BPE, 1K ctx)
Mamba-1 + Copy	132M	5B	**48.3**	Ours (GPT-2 BPE, 1K ctx, uncapped)
Mamba-2 (paper)	180M	100B	41.8	Llama-3.1 tok, 2K ctx
Mamba-3 SISO (paper)	180M	100B	37.7	Llama-3.1 tok, 2K ctx
Mamba-3 MIMO (paper)	180M	100B	32.1	Llama-3.1 tok, 2K ctx

Table 14: LAMBADA perplexity, **cross-paper context only**—not a controlled comparison. Differences in tokenizer (GPT-2 BPE vs. Llama-3.1), context (1K vs. 2K), parameters (132M vs. 180M), and training budget (20×) preclude causal comparison. The Mamba-1 + Copy row is the uncapped sweep-pipeline pass of Appendix G (the pass that gives 32.1% LAMBADA accuracy at $c = 1.0$). Published numbers are from Lahoti et al. (2026).

J Seed Variance and Capacity Controls at 45M

To estimate whether the headline effects could be explained by training-seed variance or added capacity alone, we ran a controlled study at 45M scale (6-layer Transformer, d=512, Variant B copy head, 2B FineWeb-Edu tokens). Table 15 reports two seeds per arm and three capacity controls.

K Three-Seed Replication at 60M Across All Five Families

The 45M study above varies the seed for one Transformer only. To test whether the every-pair improvement of Table 1 survives a change of training seed in every family, we ran a three-seed study at 60M scale: each of the five families with and without its copy head for 1B FineWeb-Edu tokens (drawn from the same 2B-token file) at seeds 42, 137, and 789, 30 runs in all. The runs use the 130M training scripts with 8 layers and d_{model}=512 (768 for Mamba-1; Jamba-style attention at

Model	Params	WT-103 PPL ↓	LAMBADA Acc ↑
Baseline (seed 42)	44.9M	74.3	5.0%
Baseline (seed 137)	44.9M	73.9	4.5%
+Copy (seed 42)	45.5M	59.9	19.3%
+Copy (seed 789)	45.5M	60.2	19.2%
FLOP-matched (7 layers, no copy)	48.1M	70.3	5.6%
Param-matched (wider FFN, no copy)	48.5M	72.8	4.9%
Scrambled-source copy (seed 42)	45.5M	62.2	3.5%

Table 15: **Seed variance and capacity controls at 45M scale** (Variant B, 2B tokens). Seed spread: ≤ 0.4 PPL and ≤ 0.5pp LAMBADA. Copy effect: $\sim$14 PPL and $\sim$14pp LAMBADA. The FLOP-matched and parameter-matched baselines recover at most 4.0 of the 14.4 PPL points the copy head gains (74.3 $\rightarrow$ 70.3 and 72.8) and at most 0.6 of its 14.3pp LAMBADA points, so the benefit is not added capacity. The scrambled-source control, *trained* with source identities permuted, retains $\sim$80% of the PPL gain (74.3 $\rightarrow$ 62.2) but none of the LAMBADA gain (3.5% $\approx$ baseline 5.0%). Its permutation was drawn over all L positions (`randperm`(L), redrawn every step), so the copy distribution at position t contained the target itself whenever a visible source was mapped to $t+1$ (expected mass $1/L$ per position, a stochastic floor that a trained gate can learn to exploit on uncertain tokens); the retained PPL fraction therefore mixes output-distribution expressivity (Yang et al., 2018; Chang & McCallum, 2022) with this target exposure and is not a clean estimate of either, and the causal inference-time control of Table 3 is a separate intervention that does not replace the missing training-time experiment (Section 6). Caveats: this study is at 45M, uses Variant B only, and has two seeds per arm rather than three.

2 of the 8 layers), the 130M optimizer, schedule, sequence length, gate variant, and effective batch (64×1024 tokens), d_{copy}=128 for the four heads that use 256 at 130M (d_{copy}=d_{model} for GPT-2-style, as at 130M), and the gate warmup halved to 7,500 steps so that it remains the same fraction of training. Each run is scored through the backend of Section 4 (Variant A models at the training cap $c = 0.5$, as in Table 1; their LAMBADA gains are therefore not comparable with the uncapped 2B rows of Table 6 or the uncapped points of Appendix G) on WikiText-103 (word perplexity) and LAMBADA, and on 200 fixed 1,024-token FineWeb-Edu windows from the last 2% of the training file. That region is held out from the two Transformers, whose scripts train on the first 98%, but the three SSM-family scripts sample from the whole file, so for them each window was seen about 0.5 times on average: the FineWeb column is a quasi-holdout, not the CC-MAIN-2024-51 holdout of Table 1, and WikiText-103 is the clean cross-family number. Table 16 reports the raw baseline and +copy values for all three seeds. All 60 paired cells (5 architectures $\times$ 4 metrics $\times$ 3 seeds) are positive: the copy head helps every family on every metric in every seed.

L Causality Audit

We verify that the copy head introduces no future-token leakage through a six-test audit (`scripts/scale/audit_copy_causality.py`):

1. **Future-perturbation invariance:** Perturbing tokens after position p produces exactly 0.00 log-probability difference at position p.
2. **Full-sequence vs. autoregressive equivalence:** Full-sequence logits match token-by-token autoregressive decoding (to within kernel precision, 10^{-4} uncompiled).
3. **Copy attention mask:** Copy attention at position t has zero weight on positions $j \geq t$ (self and future blocked; as noted in Section 3, blocking $j = t$ is conservative, since x_t is observed).
4. **Novel target:** Tokens never appearing in context receive exactly zero copy probability.
5. **Permuted-source negative control:** Scrambling copy source positions degrades NLL (GPT-2-style: +56%).
6. **Shift/mask consistency:** Position t's prediction targets token x_{t+1}, verified at all positions.

		FineWeb PPL ↓		WT-103 PPL ↓		LAMBADA acc. (%) ↑		LAMBADA PPL ↓	
Architecture	Seed	Base	+Copy	Base	+Copy	Base	+Copy	Base	+Copy
GPT-2-style	42	45.80	41.42	117.9	90.7	10.8	19.7	2977	225
	137	45.85	41.26	118.1	90.4	10.1	19.5	2779	228
	789	47.05	41.44	121.6	90.3	11.2	19.4	3142	222
Llama-style	42	57.04	48.75	159.9	112.6	11.9	19.8	4393	254
	137	56.03	48.21	159.7	111.8	10.9	18.9	4937	271
	789	60.46	47.79	175.4	110.1	9.8	19.5	6434	256
Mamba-1	42	32.57	30.45	84.2	66.7	14.5	20.6	443	162
	137	32.67	30.14	85.2	67.2	12.7	20.8	729	160
	789	32.49	30.42	84.6	67.6	12.7	19.7	593	171
Jamba-style	42	36.39	33.82	92.0	74.8	12.8	20.0	1625	171
	137	36.42	33.90	92.5	74.5	12.6	20.2	1490	168
	789	36.43	33.87	91.4	74.7	13.1	21.5	1486	160
GDN	42	30.15	28.15	77.8	61.6	12.1	22.1	894	136
	137	30.05	28.06	76.5	60.8	12.9	23.3	884	127
	789	30.24	28.16	77.0	60.9	11.5	21.9	1051	135

Table 16: **Three-seed replication at 60M** (1B tokens; raw values of both arms in each seed; Variant A at $c = 0.5$). The copy head improves all four metrics for all five families in all three seeds (60 of 60 baseline/+copy comparisons). FineWeb is the quasi-holdout of the text, seen ≈0.5 times during training by the three SSM-family models; WikiText-103 and LAMBADA are held out for all five. Caveats: three seeds per arm at 60M scale and 1B tokens, so this replicates the sign and rough size of the 130M effect, not its magnitude, and the Llama-style baseline varies most between seeds.

All six tests were run and passed on GPT-2-style +Copy. For Llama-style, Mamba-1, Jamba-style, and GDN+Copy, only test 1 (future-perturbation invariance) was run; it passes with exactly 0.00 difference on all four. Tests 2–6 were not run on these models.

M Reasoning and Retrieval Diagnostics (2B Tokens)

Table 17 presents results on a custom reasoning evaluation suite (cloze-format probes) at 2B tokens for the three of the five 2B pairs that were scored on it. Each task probes a different facet of in-context processing.

	GPT-2-style		Mamba-1		Jamba-style	
Task	B	+C	B	+C	B	+C
State tracking	51.5	52.0	63.5	59.5	41.0	**68.0**
Variable binding	46.0	41.5	31.0	54.0	63.5	53.0
Multi-hop*	72.5	98.0	100.0	100.0	71.5	100.0
Reversal	30.5	36.5	29.0	17.5	21.5	22.0
MQAR-update	43.5	**60.0**	11.0	15.0	47.0	16.0
Average	48.8	**57.6**	46.9	49.2	48.9	51.8

Table 17: **Reasoning diagnostics at 2B tokens** (130M scale, cloze format). B = baseline, +C = with copy head. The copy head substantially improves associative recall for GPT-2-style (MQAR-update: +16.5pp) and state tracking for Jamba-style (+27pp). Mamba-1 shows modest overall gains but a large variable-binding improvement (+23pp). *Multi-hop reaches 100% for all SSM-family +copy models and may be degenerate despite v2.1 distractor chains; interpret with caution. These are 2B-token results; the 5B results under the same suite are in Table 21, where the GPT-2-style MQAR-update gain replicates (+22pp) while Llama-style, Mamba-1, and GDN show an MQAR-update *regression*.

Key observations. (1) The MQAR-update task—which requires overwriting a previously stored key–value binding with a new value—shows the largest copy-head gain for GPT-2-style (+16.5pp), directly testing associative recall. (2) Jamba-style state tracking improves dramatically (+27pp), suggesting the copy head helps the hybrid backbone maintain entity state across context. (3) Mamba-

1's MQAR-update remains weak at 2B even with the copy head (15%); at 5B, however, the Mamba-1 baseline reaches 69% (Table 21), so this reflects the 2B budget rather than a fixed limitation of the backbone. (4) Variable binding shows mixed effects: Mamba-1 improves (+23pp) but Jamba-style and GPT-2-style degrade, possibly reflecting interference between the copy mechanism and the backbone's existing binding strategy.

The 5B checkpoints are evaluated on the same suite, plus copyability-stratified scoring and two entity probes, in Appendix N; closed-gate inference ($c = 0$) for the Variant A models is in Section 5.4 and Appendix H. RULER-style needle-in-a-haystack probes (Hsieh et al., 2024) within our 1K-token context remain future work.

N Retrieval Diagnostics at 5B Tokens

This appendix tabulates the diagnostics behind Section 5.3 and the Analysis. All were run on the 5B checkpoints of Table 1 in one pass through one scoring backend (the same copy-mixture log-probabilities as the 11-task evaluation), on a single RTX 3090, with Variant A copy models at the training cap $c = 0.5$ (and, for Section 5.4, additionally at $c = 0$), Variant B models with their softmax gate, and baselines with the gate forced to zero. The GPT-2-style 5B baseline was scored in a second pass with the same scripts after its final checkpoint was restored from backup (its one training asymmetry with the +copy run is described in Appendix A).

Copyability-stratified scoring. A scored token at position i is *copyable* if its identity occurs at some position j with $1 \leq i - j \leq 512$ (the Variant A window); the distance $d = i - j$ to the most recent occurrence is bucketed. In these target coordinates the head at position $t = i - 1$ reaches sources $j \geq t - 512$, *i.e.* $d \leq 513$, so the copyable set omits the single farthest reachable distance $d = 513$; we did not recompute the split from the exact source mask and expect the omission to be negligible. The d=1 bucket (an immediate repeat) is reported separately because the copy head attends to keys strictly before the query and cannot copy it (Section 3). LAMBADA uses the lm-eval `lambada_openai` test split (5,153 examples; an example is copyable if all its continuation tokens are); WikiText-103 uses the lm-eval document-level test split with lm-eval's detokenizer and non-overlapping 1,024-token windows, exactly as the rolling-likelihood evaluation of Table 1. The unstratified totals reproduce the lm-eval numbers to two decimals (*e.g.* Mamba-1 baseline LAMBADA perplexity 107.59 vs. 107.62).

	Copyable targets (73.7%, n=3,798)					Non-copyable targets (26.3%, n=1,355)			
	accuracy (%) ↑			PPL ↓		accuracy (%)		PPL ↓	
Architecture	Base	+Copy	Δpp	Base	+Copy	Base	+Copy	Base	+Copy
GPT-2-style TF	23.5	34.5	+11.0	159.4	20.5	2.6	1.5	3777	**5184**
Llama-style TF	28.0	39.1	+11.1	90.2	14.8	2.7	1.5	3119	**4021**
Mamba-1	30.2	34.3	+4.1	44.4	20.2	3.5	2.1	1285	**2843**
Jamba-style	28.8	36.4	+7.6	66.2	17.8	4.1	2.7	1444	**2372**
GDN	26.6	35.4	+8.8	85.2	18.4	3.9	2.6	1256	**2861**

Table 18: **LAMBADA stratified by copyability** (5B models, single seed; Section 5.3). An example is copyable if its target token occurs in the 512 tokens preceding it. Copy-head gains are confined to copyable targets; on non-copyable targets perplexity worsens in every pair (bold). Variant A models are scored at the training cap $c = 0.5$ (aggregate accuracies in this scorer 25.8%, 27.5%, 26.8% for Mamba-1, Jamba-style, GDN; Table 1). Perplexity is the lm-eval definition.

Model	Per-token PPL ↓ Copyable	Non-copy.	Copyable per-token PPL by distance d $[2, 8]$	$[9, 32]$	$[33, 128]$	$[129, 512]$	Word-PPL (lm-eval)
GPT-2-style base	9.38	185.5	4.92	6.01	9.97	24.64	61.67
GPT-2-style +Copy	7.03	176.5	4.19	4.88	7.31	15.79	48.87
Llama-style base	8.71	159.7	4.64	5.67	9.25	22.25	56.45
Llama-style +Copy	6.59	155.6	3.97	4.62	6.84	14.50	43.59
Mamba-1 base	8.57	140.9	4.74	5.46	8.70	23.89	52.78
Mamba-1 +Copy	6.95	**149.2**	4.06	4.78	7.22	16.02	43.70
Jamba-style base	8.08	135.9	4.44	5.36	8.49	20.00	49.30
Jamba-style +Copy	6.54	**140.3**	3.97	4.60	6.76	14.31	40.99
GDN base	8.46	132.7	4.56	5.47	8.78	22.53	48.44
GDN +Copy	6.71	**138.9**	3.97	4.67	6.95	15.12	41.38

Table 19: **WikiText-103 test split stratified by copyability** (5B models; per-token perplexity over 271,734 scored tokens, of which 146,897 (54.1%) are copyable at distance $d \in [2, 512]$ to the last occurrence, 1,194 (0.4%) are immediate repeats ($d = 1$, excluded from the "Copyable" column), and 123,643 (45.5%) are non-copyable). Copy heads lower perplexity on copyable tokens at every distance, most at long range; on non-copyable tokens they raise it for the three Variant A models (bold) and lower it slightly for the two Transformers. Variant A at $c = 0.5$; the word-perplexity column is the lm-eval total of the same run.

The `\n\n` artifact. Of the 1,194 immediate-repeat tokens, 1,178 are the second newline of a `\n\n` pair emitted by the GPT-2 tokenizer after each WikiText section heading. Our FineWeb-Edu training subset essentially never contains this bigram, so baselines assign it a mean NLL of 15–25 nats ($p \approx 10^{-7}$–10^{-11}), which accounts for 7–12% of their word-perplexity; copy heads copy an earlier newline in the window (6.6–10.4 nats; 3–5%). Table 20 reports word-perplexity with and without these tokens; the copy-head reduction shrinks by an eighth (GPT-2-style) to a half (Mamba-1) once they are excluded, and the relative sizes of the reductions change (GDN's, the smallest of the three Variant A reductions with the artifact included, becomes the largest without it)—one more reason not to rank families by their WikiText gain. We regard the excluded numbers as the more faithful estimate of the copy head's effect on WikiText.

Architecture	Mean NLL on `\n\n` (nats) Base	+Copy	Word-PPL, all tokens Base → +Copy	Δ%	Word-PPL, excl. `\n\n` Base → +Copy	Δ%
GPT-2-style TF	16.9	10.4	61.67 → 48.87	−20.8	56.73 → 46.42	−18.2
Llama-style TF	23.5	8.3	56.45 → 43.59	−22.8	50.25 → 41.83	−16.8
Mamba-1	25.0	6.6	52.78 → 43.70	−17.2	46.64 → 42.29	−9.3
Jamba-style	22.2	7.7	49.30 → 40.99	−16.9	44.17 → 39.47	−10.6
GDN	15.4	7.4	48.44 → 41.38	−14.6	44.89 → 39.89	−11.1

Table 20: **WikiText-103 word-perplexity with and without the 1,178 `\n\n` tokens** (5B models, diagnostic pipeline, Variant A at $c = 0.5$). "Excl." removes those tokens' NLL from the summed NLL; the lm-eval word count (241,335) is unchanged, so the excluded numbers are exact perplexities over the remaining tokens under lm-eval's normalizer. The GPT-2-style baseline total equals that of Table 1, the Jamba-style, GDN, and Mamba-1 baseline totals agree with it within 0.3% and the Llama-style baseline within 1.4% (Appendix P), and the copy-model totals equal the $c = 0.5$ entries of Table 1; the all-token Δ% for Llama-style (-22.8) and Mamba-1 (-17.2) therefore differ from Table 1 (-21.7, -17.5) by exactly this baseline-total discrepancy.

Synthetic reasoning suite at 5B. Table 21 repeats the five-task cloze suite of Appendix M on the 5B checkpoints (reversal: given "Forward: 3 5 7. Backward: 7", predict 5). Each task has 200 generated items, scored by whether the first token of the answer outscores every distractor (up to three; chance ≈25%; binomial standard error ≈3.5pp at 50%). MQAR-update presents 3–6 key–value pairs, a correction to one key, and queries that key; the superseded value is always a distractor, and we report how often it outscores the corrected one. Multi-hop (v2.1) interleaves a distractor chain so that the last-mentioned room is wrong; we report how often that room outscores the answer.

Task (accuracy, %)	GPT-2-style B	GPT-2-style +C	Llama-style B	Llama-style +C	Mamba-1 B	Mamba-1 +C	Jamba-style B	Jamba-style +C	GDN B	GDN +C
State tracking	42.5	36.5	66.0	**76.0**	52.0	**76.5**	58.5	**81.0**	56.0	**71.5**
Variable binding	42.5	41.5	44.5	40.5	38.5	40.0	58.0	62.5	42.0	44.5
Multi-hop (v2.1)	43.5	55.5	30.0	37.5	42.0	54.5	44.0	49.0	42.5	49.5
Reversal	34.5	30.0	35.0	23.0	28.5	25.5	26.5	24.5	25.5	27.5
MQAR-update	37.0	**59.0**	71.0	**62.5**	69.0	**59.0**	20.0	21.0	82.5	**28.0**
Average	40.0	44.5	49.3	47.9	46.0	51.1	41.4	47.6	49.7	44.2
Old value wins (MQAR-update, %)	46	**26**	22	24	13	26	62	62	8	**64**
Wrong room wins (multi-hop, %)	52	44	62	62	48	45	52	51	56	50

Table 21: **Synthetic reasoning suite at 5B tokens** (single seed; 200 items per task; Variant A at $c = 0.5$; two-proportion standard error ≈5pp near 50%). B = baseline, +C = with copy head. "Old value wins" is the share of MQAR-update items on which the superseded value outscores the corrected one; "wrong room wins" is the share of multi-hop items on which the last-mentioned (distractor) room outscores the answer. Bold marks the architecture-dependent effects: state tracking improves for Llama-style and the three SSM-family models (+10 to +25pp) but not for GPT-2-style (−6pp, about one standard error); MQAR-update—recall after an in-context correction—*regresses* for Llama-style, Mamba-1, and GDN (the superseded value wins 64% of GDN items with the copy head versus 8% without), is flat for Jamba-style, which already fails it at baseline (20%, superseded value wins 62%), and *improves* by 22pp for GPT-2-style (superseded value wins 26% versus 46%). Reversal falls by 12pp for Llama-style (about 2.4 standard errors); the other reversal changes are within about one standard error.

Discussion of the reasoning suite. The 5B copy heads raise state-tracking accuracy by 10–25 points for Llama-style and the three SSM-family models, but lower MQAR-update accuracy for three of them (GDN 82.5→28.0%, Mamba-1 69.0→59.0%, Llama-style 71.0→62.5%; Jamba-style already fails it at baseline), and for GDN the superseded value now wins 64% of items (8% at baseline). For the GPT-2-style Transformer the pattern inverts: the head raises MQAR-update from 37.0 to 59.0% (+22 points, about 4.5 two-proportion standard errors; the superseded value wins 26% of items versus 46%) and lowers state tracking from 42.5 to 36.5%. The GPT-2-style baseline is also weak on MQAR-update (37%, against 69–82% for three of the other four baselines; Jamba-style is lower at 20%), so there the head plausibly supplies retrieval the backbone lacks, whereas on backbones that already retract, a head that retrieves whichever value co-occurred with the key, with no mechanism for retraction, competes with them. The entity probes below are too small to interpret; the rise in GDN's training-time (vocabulary-only) perplexity is explained in Appendix T.

Entity probes. Two small probes ask whether the copy head helps reproduce multi-subword names. The first (which we call Mahānāma after one of its names) inserts each of 30 romanized Sanskrit epic names—whose IAST diacritics fragment into 2–10 GPT-2 subtokens—three times into one of eight English templates and measures the mean per-subtoken log-probability gain of the later occurrences over the first; only two names (Vidura, Bharata) fall in the "short" ($\leq$2-subtoken) bin. A second part, which tracked two names in one passage, is omitted because the aggregation behind its percentages could not be verified. The second probe, in the spirit of LitBank (Bamman et al., 2019) but *not* the LitBank corpus, scores eight short passages adapted from public-domain novels with hand-labeled character names, reporting perplexity on name tokens versus other tokens and an eight-item pronoun-antecedent cloze. Table 22 reports both. The copy head raises the reproduction gain for the two short names on every family, but leaves the 28 long names flat (-0.1 to $+0.4$ nats), so the hypothesis that longer names benefit more is not supported; name-token perplexity on the literary passages falls for Llama-style and Jamba-style, rises for Mamba-1 and GDN, and is unchanged within noise for GPT-2-style. With eight to thirty items per cell we draw no further conclusion from these probes.

Model	Sanskrit-name probe Reproduction gain (nats/subtoken) ↑ all (n=30)	short (n=2)	long (n=28)	Literary-passage probe PPL ↓ name tok.	other tok.	Antecedent acc. (of 8)
GPT-2-style base	3.51	3.33	3.53	491	54.4	5
GPT-2-style +Copy	3.58	4.21	3.54	502	47.2	5
Llama-style base	3.61	3.79	3.59	2110	52.6	6
Llama-style +Copy	3.53	4.50	3.46	1163	42.7	4
Mamba-1 base	2.99	1.98	3.06	178	37.6	6
Mamba-1 +Copy	3.03	3.79	2.98	277	36.9	5
Jamba-style base	2.68	2.68	2.68	688	38.4	4
Jamba-style +Copy	3.19	4.95	3.06	267	37.0	4
GDN base	2.94	3.12	2.92	63	37.9	7
GDN +Copy	3.19	4.48	3.09	132	36.6	4

Table 22: **Entity probes at 5B tokens** (single seed; Variant A at $c = 0.5$). Reproduction gain is the mean per-subtoken log-probability of a name's later occurrences minus its first. Both probes are small and synthetic; we report them for completeness and draw no conclusion beyond the text.

O Identity Control and Gate Statistics

Both diagnostics of this appendix were produced in one pass of the lm-eval backend over the 5B checkpoints (LAMBADA and WikiText-103; Variant A at the training cap $c = 0.5$) on a single RTX 3090.

Causal source shuffle. With `--shuffle-source causal`, source position j contributes the identity $x_{s(j)}$ to Eq. 3, where $s(j)$ is drawn uniformly from $\{0, \ldots, j\}$, once per batch, from a fixed generator (seed 1234); $s(j)$ never exceeds j, so no scored position receives a token it could not see unshuffled. Query/key attention (Eq. 2) and the gate are computed from $\mathbf{h}$ exactly as trained. Table 3 reports the result. On LAMBADA accuracy and WikiText-103 the shuffled Variant A models fall below their closed-gate backbones (Table 2); on LAMBADA perplexity alone they remain better than them (1,517–2,132 versus 2,190–3,203), because a head fed random earlier identities still acts as an in-context unigram cache. This differs from the 45M scrambled-source arm of Appendix J in two ways: that arm was *trained* with scrambled sources, and its permutation was drawn over all L positions (`randperm`(L), redrawn every step), so that at position t the copy distribution contained the target x_{t+1} whenever the position mapped to $t + 1$ was visible, with expected mass $\sum_{j<t} \alpha_{t,j}/L = 1/L$; the causal shuffle has no such path.

Gate statistics. With `--report-gate-stats`, the backend records the effective copy weight g_t ($c\,\sigma(z_t)$ for Variant A, $\pi_{\text{copy},t}$ for Variant B) and the maximum copy-attention weight $\max_j \alpha_{t,j}$ at every scored position $t \geq 1$ (425,077 LAMBADA and 271,734 WikiText-103 positions per model). Table 23 summarizes them; percentiles are read from 100-bin histograms. For the Variant A models the 99th percentile equals the cap on WikiText-103 and 4–7% of positions fall in the top $[0.49, 0.50]$ bin, so the cap binds on a minority of positions; the uncapped Variant B gates reach 0.9998 at their maximum and exceed 0.45 on 11–12% of WikiText-103 positions.

Model	Task	Effective copy weight g_t mean	p50	p90	p99	Fraction of positions $g_t{>}0.10$	$g_t{>}0.45$	Mean $\max_j \alpha_{t,j}$
GPT-2-style TF	LAMBADA	0.045	0.02	0.10	0.64	9.5%	1.6%	0.52
	WikiText-103	0.169	0.08	0.50	0.99	41.7%	11.2%	0.31
Llama-style TF	LAMBADA	0.061	0.03	0.15	0.72	14.2%	2.5%	0.53
	WikiText-103	0.181	0.09	0.52	1.00	46.0%	11.9%	0.30
Mamba-1	LAMBADA	0.054	0.02	0.15	0.50	14.7%	1.7%	0.56
	WikiText-103	0.160	0.11	0.46	0.50	51.2%	10.3%	0.35
Jamba-style	LAMBADA	0.035	0.02	0.08	0.47	7.4%	1.1%	0.54
	WikiText-103	0.117	0.06	0.38	0.50	34.5%	7.0%	0.31
GDN	LAMBADA	0.045	0.02	0.12	0.49	11.4%	1.5%	0.56
	WikiText-103	0.146	0.09	0.45	0.50	45.5%	9.8%	0.35

Table 23: **Effective copy-gate statistics at every scored position** (5B models, single seed; Variant A at $c = 0.5$, so $g_t \leq 0.5$; Variant B uncapped). The median gate is 0.02–0.11 on both tasks; the gate opens on a minority of positions, pinned at the cap for Variant A (p99 $= 0.50$ on WikiText-103) and near 1.0 for Variant B. The last column is the mean over positions of the largest single copy-attention weight.

P Evaluation Bookkeeping

The following recording asymmetries, each also noted where it arises, are collected here for reference. (1) Every 5B benchmark cell (Tables 1, 8, 11) comes from one backend revision and one metric per task (`acc_norm` for HellaSwag, PIQA, ARC, OpenBookQA, and SciQ; `acc` otherwise). An earlier revision of these tables mixed `acc` and `acc_norm` within a few pairs and scored the Jamba-style and GDN baselines with an untrained gate active; the apparent regressions it showed (a 14-point GPT-2-style OpenBookQA drop, 7–8-point Jamba-style ARC-E and SciQ and GDN SciQ drops) were metric mismatches and do not exist under a single metric. (2) The Variant A baselines at 2B (Appendix D) were recorded with that earlier backend revision, in which an untrained gate contributed ≈2–5% of the mixture; at 5B the same revision overstated the Jamba-style and GDN baseline WikiText-103 perplexities by 12% and 9% (55.14 and 52.58 against 49.35 and 48.45), the Llama-style baseline by 1.8% (56.67 vs. 55.68), and Mamba-1's by 0.5% (53.23 vs. 52.94). (3) True-holdout perplexity (Table 1) was measured for the five 5B pairs but not for the 2B runs, including the Mamba-3 SISO pair. (4) The benchmark tables report no confidence intervals, paired tests, or multiplicity corrections (the probe table, Table 29, reports bootstrap intervals); the binomial standard error of LAMBADA accuracy at these levels is about 0.6 points (5,153 examples). (5) Cap values (Appendix G) were examined on the benchmarks rather than selected on validation data; all 5B Variant A numbers use the training cap $c = 0.5$, and the uncapped point appears only in the sweep. (6) The GPT-2-style 5B baseline and +copy runs used different micro-batch/accumulation splits, so their learning-rate warmups differ in optimizer steps (Appendix A). (7) Exceptions to the $c = 0.5$ protocol: the 2B runs (Appendix D) and the 350M Mamba-1 pair (Appendix E) were scored only uncapped; the 2B Mamba-3 SISO pair was scored only on LAMBADA and WikiText-103, at all five caps (Table 13); an earlier evaluation of the Mamba-3 pair through a separate script, and the training scripts' vocabulary-only perplexities, are superseded by the unified backend (Section 5.6, Appendix T). (8) Mamba-1's LAMBADA accuracy at $c = 0.5$ is 25.9% in the 11-task pass (Tables 1, 8), 26.0% in the identity-control pass (Table 3), and 25.8% in the stratified scorer (Table 18), a 0.2pp spread (about ten of 5,153 examples) that we provisionally attribute to non-deterministic bfloat16 kernels and differences between scoring paths but have not traced example by example; Jamba-style and GDN agree to 0.1pp across passes. The stratified scorer's WikiText-103 totals (Tables 19, 20) agree with the lm-eval totals of Table 1 within 0.3%, except the Llama-style baseline (56.45 vs. 55.68, 1.4%; both passes force the gate to zero, and we have not traced the discrepancy). (9) Latency (Section 3) was measured in a single pass per model on one RTX 3090 (`torch` 2.4, median of 20 timed forwards after five warmups, bfloat16 autocast for the backbone and float32 for the copy mixture) and not repeated; the copy attention takes 0.32–0.53 ms against 13.9–53.3 ms for the backbone; the dense $L \times |V|$ copy distribution that the scoring backend materializes adds a further 13–90% and 1.0–1.7 GB, a scoring artifact, since a decoder scatters at most W attention weights into one vocabulary vector per step. (10) The 45M scrambled-source control drew its permutation over all L positions (Appendix O); we have not re-run it with a causal permutation. (11) The closed-gate

Transformers (Table 2) were scored through the lm-eval backend on all eleven benchmarks but not with the copyability-stratified scorer. (12) The GPT-2-style Variant A pair (Appendix R) was scored at $c = 0.5$ only, without the true-holdout, copyability-stratified, or gate-statistics passes; its closed-gate run covers four benchmarks and its shuffled-source LAMBADA perplexity was not recorded; its PIQA, ARC-E, and SciQ cells are `acc`, unlike Table 8. (13) The 60M replication (Appendix K) uses three seeds; the 350M HellaSwag, ARC-E, and SciQ cells (Table 7) were recorded from the 350M logs' default metric and are not guaranteed to match the metric convention of Table 8.

Table 24 maps every results table to the checkpoints, scorer, backend revision, inference-time cap, and seeds behind it.

Table(s)	Checkpoints	Scorer	Backend revision	Var. A cap	Seed(s)
Tables 1, 8, the Base/Full rows of 11, and the Base/Full LAMBADA and WT-103 columns of 2	130M, 5B	lm-eval backend, 11 tasks; true-holdout scorer	current	0.5	42
Closed rows of Tables 2 and 11; Figure 1	130M, 5B	lm-eval backend with the copy pathway removed	current	0	42
Tables 18, 19, 20; split columns of Table 2	130M, 5B	copyability-stratified scorer (same log-probabilities as the backend)	current	0.5 (and 0 for the split columns)	42
Tables 3, 23	130M, 5B	lm-eval backend, `--shuffle-source causal` / `--report-gate-stats`	current	0.5	42
Tables 9, 10; Figure 3	130M Var. A, 5B	sweep pipeline (lm-eval backend)	current; an earlier revision gave 48.00 / 43.49 at $c = 1.0$	0.1–1.0	42
Table 25	130M Var. B, 5B	lm-eval backend, copy window truncated to $W = 512$ at inference	current	—	42
Table 26; last rows of Tables 2, 3	GPT-2-style Var. A, 5B	lm-eval backend (`acc` for PIQA, ARC-E, SciQ)	current	0.5 / 0 / shuffled	42
Tables 21, 22	130M, 5B	synthetic cloze suite and entity probes	—	0.5	42
Table 6	130M, 2B	lm-eval backend	Var. A baselines: earlier revision (untrained gate active)	1.0	42
Table 17	130M, 2B	synthetic cloze suite (v2.1)	—	not recorded	42
Table 7; Figure 2	350M, 5B	lm-eval backend (WT-103, LAMBADA); HellaSwag, ARC-E, SciQ from the 350M logs' default metric	current	Mamba-1: 1.0	42
Tables 12, 13	2B	lm-eval backend	current (Mamba-1 baseline: earlier revision)	0.1–1.0; Mamba-1: 1.0	42
Table 14	5B Mamba-1; published	sweep pipeline pass; Lahoti et al. (2026)	mixed	1.0	—
Table 15	45M, 2B	not recorded in this paper	—	—	42, 137 (base); 42, 789 (+copy)
Table 16	60M, 1B	lm-eval backend; 200-window quasi-holdout	current	0.5	42, 137, 789
Appendix U	130M TF baselines, 5B	standalone neural-cache scorer	—	—	42

Table 24: **Evaluation provenance of every results table.** "Current" is the backend revision of Section 4 (baselines scored with the gate forced to zero; one metric per task); "earlier revision" is the superseded one described in item (2). "—" marks a cell that does not apply.

Q Span Probe: Truncating the Transformer Copy Window at Inference

To measure how much of the Variant B heads' gain depends on their full-prefix span, we re-score the trained 5B GPT-2-style and Llama-style copy models through the backend of Table 1 with the copy window truncated to the Variant A span, $\mathcal{W}(t) = \{\max(0, t - 512), \ldots, t - 1\}$, at inference; weights are unchanged. Table 25 reports the result; "gain retained" is the truncated model's improve-

ment over the baseline divided by the full-prefix model's. LAMBADA accuracy and perplexity are unchanged (the latter to four decimals: 87.6712 and 64.5497), so no scored LAMBADA target depends on a source more than 512 tokens back. Because the heads were trained on the full prefix, truncation introduces a train–test mismatch (attention mass that was placed beyond the window is renormalized over the remainder); a head trained with $W = 512$ would plausibly retain at least this much, because it would never have learned to place mass beyond the window in the first place, so the retained fractions are conservative. A GPT-2-style model trained from scratch with the complete Variant A head (sigmoid gate, $c_{\max} = 0.5$, $W = 512$, $d_{\text{copy}} = 256$, micro-step warmup), the direct test of the remaining gate-form and warmup confound, retains 74% of the WikiText-103 gain and 68% of the LAMBADA gain (Appendix R).

Model	Metric	Baseline	Full prefix	$W = 512$	Gain retained
GPT-2-style TF	FineWeb holdout PPL ↓	21.99	19.81	19.92	95%
GPT-2-style TF	WikiText-103 PPL ↓	61.67	48.87	50.14	90%
GPT-2-style TF	LAMBADA acc. (%) ↑	18.0	25.8	25.8	100%
Llama-style TF	FineWeb holdout PPL ↓	19.70	18.09	18.19	94%
Llama-style TF	WikiText-103 PPL ↓	55.68	43.59	44.71	91%
Llama-style TF	LAMBADA acc. (%) ↑	21.3	29.2	29.2	100%

Table 25: **Span probe** (5B Variant B copy models, single seed). The trained full-prefix heads are re-scored with the copy window truncated to $W = 512$ at inference. Truncation costs 5–10% of the WikiText-103 and holdout gains and none of the LAMBADA gain; with span equalized, the holdout reductions are -9.4% (GPT-2-style) and -7.7% (Llama-style), versus -3.3 to -4.2% for the Variant A models at $c = 0.5$. Baselines and full-prefix values are those of Table 1.

R GPT-2-STYLE TRANSFORMER WITH THE VARIANT A HEAD

To remove the gate-form, warmup, and d_{copy} confound of Section 3 for one backbone, we retrained the GPT-2-style model from scratch at 5B tokens with the complete Variant A head of the SSM-family runs: $d_{\text{copy}} = 256$, $W = 512$, scalar sigmoid gate with $b_g = -3$, $c_{\max} = 0.5$, $N_{\text{ramp}} = 15$K micro-steps (394K parameters on the 163.0M backbone), on the same data, seed, and schedule as its baseline and Variant B partner (Table 4), with micro-batch 16 × accumulation 4—the baseline's split, so this pair has none of the learning-rate-warmup asymmetry of the Variant B pair, and the gate warmup spans ≈123M tokens as for Mamba-1. The model was scored through the backend of Table 1 at the training cap $c = 0.5$ on all eleven benchmarks, with the gate closed ($c = 0$; LAMBADA, WikiText-103, HellaSwag, SciQ), and under the causal source shuffle of Appendix O (LAMBADA accuracy and WikiText-103); the true-holdout and copyability-stratified scorers and the gate statistics were not run. Table 26 reports the result next to the Variant B model of Table 8.

Three things follow. (1) *The head helps under either gate form.* WikiText-103 perplexity falls 15.3% and LAMBADA accuracy rises 5.3pp (perplexity 366 → 101), against -20.8% and $+7.8$pp for the Variant B head on the same backbone: the Variant A head retains 74% and 68% of the Variant B gains, and the remainder reflects combined changes in head configuration and training recipe (uncapped two-layer gate, $d_{\text{copy}} = 768$, full-prefix span, an $8\times$ longer gate warmup, and a different learning-rate warmup), of which span alone is worth 5–10% (Appendix Q). (2) *The mechanism is the same.* Randomizing the retrieved identities drops the model to 2.9% LAMBADA and 76.8 WikiText-103, and closing the gate to 4.8% and 72.9, both below the 18.0% / 61.7 baseline and in the order shuffled < closed < baseline < full of the three SSM-family Variant A pairs (Section 6); with the gate closed HellaSwag is unchanged (29.4 vs. 30.4) while SciQ falls to 66.6 (full 79.7), as in Table 11. (3) *With the head equalized, the Transformer is not ahead.* Its gains sit inside the SSM-family Variant A range at the same cap (WikiText-103 -14.6 to -17.5%, LAMBADA $+2.6$ to $+6.1$pp; Section 5.1), so the Transformers-first WikiText ordering of Table 1 does not survive head equalization; whether the true-holdout ordering does is untested. The other nine benchmarks move as in the Variant B pair: BoolQ ($+6.9$) and SciQ ($+5.5$) rise, OpenBookQA falls 3.0 points (-0.4 for the Variant B pair), and the rest are within 1.5 points. One seed; the Llama-style arm was not trained.

Benchmark	Base	+Copy, Var. A (c=0.5) value	Δ	Var. A controls Closed	Shuffled	+Copy, Var. B (Table 8)
WT-103 PPL↓	61.7	52.3	−15.3%	**72.9**	**76.8**	48.9
LAMBADA acc.↑	18.0	23.3	+5.3	**4.8**	**2.9**	25.8
LAMBADA PPL↓	366.4	100.9	−72.5%	—	—	87.7
SciQ↑	74.2	79.7	+5.5	**66.6**	—	80.8
HellaSwag↑	29.6†	30.4†	+0.8	29.4†	—	30.7†
PIQA↑	61.2	61.9	+0.7	—	—	62.6
ARC-E↑	48.7	50.0	+1.3	—	—	51.8
ARC-C↑	23.0†	23.2†	+0.2	—	—	25.0†
WinoGrande↑	50.4	51.8	+1.4	—	—	51.6
BoolQ↑	54.9	61.8	+6.9	—	—	61.2
OBQA↑	31.6†	28.6†	−3.0	—	—	31.2†
COPA↑	65.0	66.0	+1.0	—	—	63.0

Table 26: **GPT-2-style backbone retrained with the Variant A head** (5B tokens, single seed; lm-eval-harness 0-shot). Base = the GPT-2-style baseline of Table 1; value = the Variant A model at its training cap $c = 0.5$; Δ = its change over Base (percent for perplexities, points for accuracies, computed before rounding); Closed = the same checkpoint with $c = 0$; Shuffled = with the causal source shuffle of Appendix O; the last column is the Variant B model of Table 8. Bold: every control falls below baseline. † = `acc_norm`; other accuracies are `acc`, so the PIQA, ARC-E, and SciQ rows of this table use `acc` where Table 8 uses `acc_norm` (the same baseline scores 60.6, 44.8, and 65.3 there); every column of this table is single-metric within each row. Cells marked — were not run.

S Decode Cost in the Sub-Billion Regime

Models of 130M–350M parameters are the size class deployed on phones and embedded devices, where per-token decode cost and cache memory matter more than prefill throughput. Table 27 estimates, from the configurations of Table 5, the per-token matrix-multiply FLOPs of each backbone at context T=1024 and of its copy head (query and key projections, W key scores, softmax, gate, and vocabulary mixture), together with the per-sequence cache each needs in float32. The head adds 0.3–1.5% of backbone FLOPs in every family. Its cache adds 1–6% for the Transformers, the Jamba-style hybrid, and Mamba-3 SISO, but 20–28% for Mamba-1 and GDN, whose fixed recurrent state is only 2–3 MB; for these two backbones the head's 0.5 MB key window is a real memory cost.

Architecture	Backbone GFLOP/tok	Head MFLOP/tok	Head share	Backbone cache	Head cache
GPT-2-style	0.29	4.4	1.5%	72.0 MB	3.0 MB (4.2%)
Llama-style	0.40	1.8	0.4%	72.0 MB	1.0 MB (1.4%)
Mamba-1	0.26	1.5	0.6%	1.8 MB	0.5 MB (28%)
Jamba-style	0.28	1.2	0.4%	13.1 MB	0.5 MB (3.8%)
GDN	0.40	1.2	0.3%	2.6 MB	0.5 MB (20%)
Mamba-3 SISO	0.25	1.2	0.5%	9.0 MB	0.5 MB (5.6%)

Table 27: Analytic per-token decode FLOPs and per-sequence float32 cache at T=1024.

As a measured check, we timed one decode step on a 4-core x86 CPU (PyTorch, float32). A single-token forward of the GPT-2-style backbone, which reads all weights once and is a lower bound on its decode step, takes 43.9 ms with one thread and 49.7 ms with four. One head step, with the copy mass added sparsely at the $\leq W$ window tokens to the vocabulary distribution the backbone already produces, takes 0.63–0.64 ms for the Variant B GPT-2-style head and 0.18–0.20 ms for a Variant A head (d_{copy}=256, W=512), or 1.3–1.4% and 0.4% of the backbone step. The measured share agrees with the analytic FLOP share. To time a real cached decode step for more than one backbone, we instantiated Hugging Face implementations with the configurations of Table 5 and random weights (latency does not depend on weight values) and measured one single-token step at context 512, with the KV or recurrent-state cache populated. Table 28 reports the result on the 4-core CPU above, on a desktop i5-13400F CPU, and on a consumer RTX 3060 GPU in two machines. On both CPUs the head step is 0.2–0.9% of the backbone step. On the GPU the head takes an almost constant time regardless of model size, because it is bound by kernel-launch overhead rather than arithmetic: the

same RTX 3060 model spends 0.21–0.24 ms on it behind a weak host CPU and 0.07–0.08 ms behind the i5-13400F, or 1.3–4.2% of the backbone step. Fusing the head into one kernel would lower this further. The Mamba-1 and Jamba-style backbones ran on the reference PyTorch path rather than the optimized `mamba-ssm` kernels, which inflates their backbone times and understates the head's share. GDN and Mamba-3 SISO have no Hugging Face implementation and were not timed.

	CPU (4 threads)		RTX 3060 GPU	
Architecture	4-core laptop-class backbone / head	i5-13400F backbone / head	weak host backbone / head	i5-13400F host backbone / head
GPT-2-style	53.9 / 0.51 (0.9%)	16.9 / 0.09 (0.5%)	5.6 / 0.24 (4.2%)	3.4 / 0.08 (2.3%)
Llama-style	118.3 / 0.63 (0.5%)	22.1 / 0.07 (0.3%)	9.2 / 0.24 (2.6%)	4.6 / 0.08 (1.8%)
Mamba-1	55.6 / 0.26 (0.5%)	16.2 / 0.06 (0.3%)	9.9 / 0.21 (2.1%)	4.2 / 0.07 (1.7%)
Jamba-style	79.3 / 0.20 (0.2%)	21.1 / 0.06 (0.3%)	14.2 / 0.21 (1.5%)	5.6 / 0.07 (1.3%)

Table 28: Measured single-token decode step in ms, backbone / copy head (head share), context 512, float32.

For a dedicated accelerator (ASIC or FPGA), the relevant costs are data movement and on-chip memory. We estimate them from the energy figures of Horowitz (2014): 1–2 nJ per 64-bit DRAM access against roughly 10–20 pJ per on-chip access. At 130M–200M parameters in float32 the weights do not fit on chip and are streamed from DRAM at every decode step, which costs 62–198 mJ per token across the six backbones and dominates all arithmetic. The head adds only its own weights and a 512-key window, 0.2–0.5% of that energy for the four SSM-family backbones and 0.3–1.1% for the two Transformers (whether the window is kept on chip or in DRAM changes this by at most 0.7 points). The binding constraint is on-chip memory instead. An SSM accelerator can hold the whole recurrent state on chip, and the head's window then enlarges that on-chip buffer from 1.8 to 2.3 MB for Mamba-1 (+28%) and from 2.6 to 3.1 MB for GDN (+20%), but only by 4–6% for Jamba-style and Mamba-3 SISO. On an FPGA, where block RAM is the scarce resource, the same window either competes with the recurrent state or moves to DRAM at the energy cost above. To test this, we implemented the head's attention step (scores over the 512-key window at $d_{\text{copy}} = 256$ and the softmax, with 16-bit fixed-point keys and queries) in VHDL for a Lattice ECP5-85F using open-source synthesis and place-and-route. With 16 parallel multiply–accumulate lanes the design routes at about 81 MHz and takes 114 μs per token (32 μs with 64 lanes, post-route estimates rather than timing sign-off); it matches a fixed-point reference bit for bit on 20 synthetic cases, and its buffers alone occupy 60% of the device's block RAM, which confirms that on-chip memory rather than arithmetic is the constraint. The projections, gate, and vocabulary mixing were not implemented in hardware. The remaining figures are estimates, not measurements: we did not run on phone hardware, measure energy, or test quantized weights, whose interaction with the float32 copy mixture of Appendix B is untested.

T Additional Analysis

The cross-architecture ordering is observational. Read across pairs, the two Transformers show the larger WikiText-103, LAMBADA, and true-holdout gains at the training cap, whereas uncapped the three SSM-family models show the larger LAMBADA gains (Table 1, Appendix G). Section 6 explains why we do not interpret this ordering. Of the copy-head differences that change with architecture class, span is measured to be minor (Appendix Q); the gate form (uncapped softmax MLP vs. capped sigmoid), the warmup horizon, and d_{copy} are separated from the backbone only for the GPT-2-style arm of Appendix R. Candidate backbone-side contributors are (1) output-distribution expressivity, which the 45M scrambled-source control suggested but which its non-causal permutation overstates, and which the causal control of Table 3 bounds without excluding: none of the gain survives identity corruption at 130M, but an inference-time corruption cannot separate expressivity from retrieval within a component whose support is the context (Section 6); (2) the backbone's capacity to *integrate* an external retrieval signal; and (3) whether the backbone already retrieves natively. The within-family Mamba-1 vs. Mamba-3 SISO comparison (Section 5.6) is the one fixed-head test of (3): the better native retriever gains as much on WikiText-103 and less on LAMBADA accuracy, so (3) at most modulates the benefit and does not remove it; the holdout ordering, in which the two backbones with full attention gain most, points the same way but is confounded by the head.

Neither settles the question. The decisive experiment is the Variant A head on every backbone with multiple seeds; its GPT-2-style arm (Appendix R) places that backbone's WikiText-103 and LAMBADA gains inside the SSM-family range at the same cap, so the ordering above is at least in part a property of the head.

U Neural-Cache Baseline on the Transformer 5B Checkpoints

To test whether a training-free cache recovers the trained copy head's gains, we evaluate a continuous neural cache (Grave et al., 2017) on the GPT-2-style and Llama-style 5B baseline checkpoints through a standalone scorer (not lm-eval-harness; absolute values differ from Table 1 because lm-eval's text detokenization and multi-token-word LAMBADA protocol are not replicated, but the same scorer evaluates both the cached and uncached models). For each position t, the cache distribution $p_{\text{cache}}(w \mid t) \propto \sum_{j \in \mathcal{W}(t): x_{j+1}=w} \exp(\theta \cdot \cos(\mathbf{h}_t, \mathbf{h}_j))$ pairs each past state $\mathbf{h}_j$ with the token x_{j+1} that followed it, as in the original cache, and mixes with the vocabulary distribution at weight λ: $p(w) = (1-\lambda)\, p_{\text{vocab}}(w) + \lambda\, p_{\text{cache}}(w)$, using the backbone's hidden states as keys with no learned projections and a 512-token window matching the Variant A copy head. We also score a variant that pairs $\mathbf{h}_j$ with its own input token x_j, as our head does. Both are computed from one forward pass per window with cosine similarity on both benchmarks.

Sweeping $\lambda \in \{0.05, 0.1, 0.2, 0.3\}$ and $\theta \in \{1, 5, 10, 20\}$: on GPT-2-style, the neural cache's best WikiText-103 reduction is 20.3% (at λ=0.1, θ=10; baseline 94.7 $\rightarrow$ 75.5 under this scorer) and its best LAMBADA accuracy 35.8% ($+0.8$ points over 35.0%); on Llama-style, the best WikiText-103 reduction is 17.1% (80.9 $\rightarrow$ 67.1) and the best LAMBADA accuracy 39.0% ($+1.3$ points over 37.7%). The own-token variant reaches only 11.8% and 8.5% on WikiText-103 and at most $+0.1$ points on LAMBADA. Scoring the trained copy models through the same standalone scorer gives WikiText-103 word perplexity 56.1 (-40.8%; GPT-2-style) and 50.0 (-38.2%; Llama-style) and LAMBADA accuracy 41.2% and 43.8% ($+6.2$ and $+6.1$ points over the same baselines, all from the current scorer), so the neural cache recovers about half of the perplexity gain and 13–21% of the accuracy gain. The trained head's advantage is thus established within one scorer: learned query/key projections and a context-dependent gate produce retrieval that a post-hoc similarity cache only partly replicates, most clearly on LAMBADA.

The own-token variant is the copy head with $W_q = W_k = I$ (identity projections, up to normalization) and a constant gate; the original cache differs from it only in pairing each state with the following token. Joint training provides at least two things a post-hoc cache does not: (i) learned query/key projections that select sources more selectively than raw hidden-state cosine similarity, and (ii) a context-dependent gate that lets the backbone offload copying at specific positions rather than everywhere, producing the output-pathway specialization of Section 5.4.

Training-time perplexity is vocabulary-only. The training scripts' logged validation perplexities are computed from the backbone's vocabulary logits alone, without the copy mixture, so for a copy model they score the closed-gate backbone of Section 5.4 on training-data windows. They accordingly *rise* with the head even as every benchmark improves—GDN 13.8 $\rightarrow$ 17.7 ($+28\%$, against $+29\%$ for its closed-gate WikiText-103 perplexity in Table 2) and Mamba-3 SISO 17.1 $\rightarrow$ 18.5—which an earlier reading took for interference with GDN's delta-rule memory. It is output-pathway offloading measured during training; whether joint training additionally interferes with a backbone's native retrieval (Table 21) is not separable from offloading with these logs.

V Hidden-State Probe of Output-Pathway Offloading

Closing the gate shows that the +copy backbone's vocabulary readout no longer predicts in-context repetitions (Section 5.4). To test whether the backbone's representation has also lost that information, we probe the final hidden state h_t, which feeds both the LM head and the copy head and is identical with the gate open or closed, of the GPT-2-style and Llama-style 5B pairs. We take 180 non-overlapping 512-token windows of WikiText-103 test, fit on 120 and test on 60 (split by window), and score positions $t \geq 32$. At each position the candidate set is the unique tokens of the prior context plus the target; a position is copyable if its target occurs in the prior context. The *native* readout ranks candidates by the backbone's own LM-head logits (the closed-gate output pathway).

The *probe* is a ridge regression from standardized h_t to a fixed random 1024-dimensional unit code per token identity, identical for both arms, with candidates ranked by cosine similarity; the ridge strength is chosen per model on a validation split of the fitting windows. Table 29 reports top-1 accuracy with 95% intervals from a paired bootstrap over test windows.

		Copyable targets (15,641)			Non-copyable targets (13,099)		
Pair	Readout	Base	+copy	Δ [95% CI]	Base	+copy	Δ [95% CI]
GPT-2-style	LM head	50.4	43.2	−7.2 [−8.3, −6.2]	21.2	25.5	+4.3 [+3.8, +4.8]
GPT-2-style	Linear probe	39.7	40.0	+0.3 [−0.2, +0.7]	6.6	6.9	+0.3 [+0.0, +0.5]
Llama-style	LM head	51.7	43.1	−8.7 [−9.8, −7.6]	22.6	27.7	+5.2 [+4.7, +5.7]
Llama-style	Linear probe	40.4	41.4	+1.0 [+0.7, +1.4]	6.7	7.5	+0.8 [+0.5, +1.0]

Table 29: **Hidden-state probe** (5B, single seed; top-1 accuracy (%) among the candidate set).

The backbone's own readout loses 7–9 points on copyable targets and gains 4–5 on the rest, mirroring Table 2, while the linear probe decodes copyable targets from the +copy hidden state at least as well as from the baseline's. Under this probe, offloading removes copying from the vocabulary readout without a measurable loss in the representation, which the copy head itself reads; the probe cannot show that the information behind the baseline's extra native accuracy is preserved. The probe is weaker than the native readout, covers the two Transformer pairs only (the SSM-family backbones need GPU kernels we did not run for this analysis), and is single-seed; a nonlinear probe could separate the arms where a linear one does not.

W Closed-Gate Replication Across 60M Seeds

To test whether output-pathway offloading replicates across training seeds, we score every available 60M +copy checkpoint of Appendix K with the copy pathway removed at inference, through the same backend and scorer as that study (WikiText-103 and LAMBADA; 14 runs, since the GDN seed-42 +copy checkpoint was not retained). Table 30 gives the range over seeds. In all 14 runs the closed-gate model falls below the independently trained baseline of the same seed on both LAMBADA accuracy and WikiText-103 perplexity.

	LAMBADA acc. (%) ↑			WikiText-103 PPL ↓		
Family (seeds)	Base	Closed	Full	Base	Closed	Full
GPT-2-style (3)	10.1–11.2	**0.5**	19.4–19.7	118–122	**176–178**	90–91
Llama-style (3)	9.8–11.9	**0.0–0.1**	18.9–19.8	160–175	**261–267**	110–113
Mamba-1 (3)	12.7–14.5	**1.1–1.3**	19.7–20.8	84–85	**123**	67–68
Jamba-style (3)	12.6–13.1	**1.1–1.4**	20.0–21.5	91–92	**119**	74–75
GDN (2)	11.5–12.9	**1.1–1.2**	21.9–23.3	77	**103–104**	61

Table 30: **Closing the copy gate at 60M, three seeds** (1B tokens; range over seeds). Bold: closed-gate worse than baseline in every seed.