RECOMPOSE: DIAGNOSING AND IMPROVING CONVERSION TO SPARSE TRANSFORMER–SSM HYBRIDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Converting a pretrained Transformer into a sparse hybrid replaces most attention modules with state-space models (SSMs), retaining only a few attention layers. We present **ReCompose**, a diagnosis-guided approach to recovering capabilities lost during conversion. We examine how replacement supervision, attention-layer selection, and joint training affect capability loss and recovery within existing conversion pipelines. The repair trains SSM replacements and their output projections to match complete teacher-layer outputs. It then jointly trains consecutive student layers to match teacher block outputs before whole-model distillation. The local and blockwise stages share a design choice: match the output passed onward while allowing the trained components to adapt within that computation. Controlled replacement experiments support the combined local repair through improved downstream accuracy. In a same-start Qwen comparison at matched post-assembly data exposure, blockwise-then-global training yields 0.6–1.0 percentage points higher MMLU accuracy, while direct global distillation yields higher Lambada accuracy at the tested endpoints. MMLU-selected layouts retain higher MMLU accuracy and weaker retrieval than retrieval-selected layouts. Subsequent training partly recovers lost capabilities with the selected attention layers held fixed. On Qwen and Llama, ReCompose recovers performance across knowledge, commonsense, and text-completion benchmarks while retaining only four attention layers. These results inform how to supervise replacements, select retained layers, and allocate training to recover the intended capabilities.

1 INTRODUCTION

Transformers use attention to access context (Vaswani et al., 2017), while state-space models (SSMs) process sequences through recurrent states (Gu et al., 2022; Gu & Dao, 2024). Combining these computations in a hybrid offers a route to more efficient sequence processing (Appendix K). Sparse conversion builds a hybrid from a pretrained Transformer by retaining a few attention layers and replacing the remaining attention modules with SSMs (Bick et al., 2024; Wang et al., 2024). The original Transformer is the *teacher*; the converted hybrid is the *student*. Selecting retained layers determines the student's structure. Replacing attention with SSMs changes how the model processes information and can impair capabilities inherited from the teacher. Recovering these capabilities is a central challenge of conversion.

A common conversion strategy trains replacements separately before refining the assembled model. MOHAWK (Bick et al., 2024) and Llamba (Bick et al., 2025a) align operations that mix information across tokens, fit intermediate outputs, and then distill whole-model predictions. Architecture-aware pipelines also combine selection and recovery (Gu et al., 2025; Yang et al., 2025). We examine how choices within these pipelines affect capability loss and recovery. Locally, which teacher output should a replacement reproduce, and should its output projection adapt? Across layers, how should independently prepared replacements train to operate on one another's outputs? At construction, which capabilities do different attention-layer selections preserve, and what can subsequent training recover without changing those selections? Controlled comparisons and recovery experiments address these questions under a fixed attention budget.

For local diagnosis, the complete decoder-layer output provides a reference: it is the hidden representation passed to the next layer. An SSM's output projection, or *readout*, maps its features into

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.

the representation used by the surrounding layer. Residual addition and the feed-forward network then transform this representation, so a smaller error at the replaced attention output need not mean a smaller error at the complete layer output. Our controlled experiments vary readout trainability and the supervised output. Learning the readout improves reproduction of both teacher outputs, whereas switching targets shifts which output is reproduced more accurately. These findings motivate adapting the readout and supervising the complete layer output; a separate task comparison supports their combined effect.

Across layers, independent fitting supplies each replacement with teacher inputs, while the assembled hybrid supplies outputs from preceding student layers. Training layers jointly covers the computation they actually perform together. Matching only a group's final output also allows its layers to compensate for one another's errors without exact imitation at every layer. This motivates joint training of consecutive layers. Whole-model distillation also permits such compensation, so the value of first training smaller groups requires an empirical comparison.

ReCompose turns these observations into a repair procedure (Figure 1). It prepares replacements with adapted readouts and complete-layer supervision. After attention selection and assembly, consecutive hybrid layers train on one another's outputs to match the teacher at each group's final output. Whole-model distillation then trains interactions across groups and the student's final predictions. The selected attention positions stay fixed during these joint-training stages.

Experiments on Qwen2.5-1.5B (Qwen et al., 2024) and Llama-3.1-8B-Instruct (Dubey et al., 2024) show recovery across knowledge, commonsense and text completion with four retained attention layers. From a common Qwen starting model, training consecutive groups before whole-model distillation improves MMLU (Hendrycks et al., 2021) by 0.6–1.0 percentage points over direct distillation at matched post-assembly data exposure. Construction experiments reveal capability trade-offs among selected layouts. Llama layer controls further show that a layer important for MMLU need not restore long-context retrieval. Subsequent Qwen training recovers some capability losses while keeping the selected attention positions fixed. Stage and objective choices also change which tasks improve. These findings support choosing layouts, objectives and checkpoints by both initial quality and recovered capabilities, using tests that vary evidence position, introduce single-error answer alternatives, and swap facts in the context. Our contributions are:

- **Diagnosis.** A controlled factorial shows that learning the SSM output transformation improves both attention-output and complete-layer fitting, whereas changing the supervised output shifts error between them. A task comparison supports the combined local repair.
- **Repair.** ReCompose combines local readout adaptation and complete-layer supervision with joint training of consecutive hybrid layers, followed by whole-model distillation. A same-start Qwen comparison favors blockwise-first training for MMLU and direct global training for Lambada.
- **Capability recovery.** Attention-layer selection produces capability trade-offs, and subsequent training can recover some losses without changing the selected layers. Under a common repair protocol, selected layouts retain distinct task profiles, while some relative rankings change during recovery.

2 Related Work

Pretrained model conversion. Transformer–SSM hybrids can be pretrained (Lenz et al., 2025) or converted from pretrained Transformers. MOHAWK (Bick et al., 2024) aligns mixing matrices, independently fits attention/SSM mixing-block outputs on teacher inputs, and distills end-to-end predictions. Its ablations support staged over direct distillation. Llamba scales this approach (Bick et al., 2025a); Mamba-in-the-Llama and RADLADS also transfer weights and distill converted models (Wang et al., 2024; Goldstein et al., 2025).

Local supervision choices. Intermediate-output matching is established (Jiao et al., 2020). LoLCATs fits pre-projection attention outputs before low-rank projection adaptation (Zhang et al., 2025); Taylor-Calibrate fits projected outputs (Zhou et al., 2026). HyLo adds a mixer target to complete-layer supervision and reports downstream gains (Ashrafi Fashi et al., 2026). Zebra-Llama converts Transformers to Mamba2 and multi-head latent attention (MLA) hybrids (Yang et al., 2025). Its released local trainer uses teacher inputs and both mixer- and layer-output losses. MLA compresses attention,

whereas we retain full attention. Within these established approaches, our factorial comparison separates readout trainability from target choice; separate downstream tests assess their combined repair and dual-target supervision (Section 4).

Training layers jointly. Block reconstruction for quantization supervises combined outputs (Li et al., 2021). Puzzle independently distills candidate decoder-layer blocks, then repairs their compatibility through end-to-end distillation after selection (Bercovich et al., 2025). Super Apriel supervises layer outputs and predictions in sampled hybrids while freezing shared feed-forward and normalization parameters (SLAM Labs et al., 2026). From one teacher input per group, ReCompose propagates student outputs through consecutive complete layers and supervises only the group's final output. All components in the active Qwen group adapt. From the same repaired assembly, we test whether allocating training to these groups before global distillation improves recovery over direct global training at matched exposure (Section 6.2).

Attention selection and capability recovery. Jet-Nemotron searches placements and replacement designs with frozen MLPs, then trains the selected model (Gu et al., 2025). Starting from Qwen2.5-1.5B, its 2B hybrid combines JetBlock with full attention placed using retrieval and sliding-window attention placed using MMLU. Gather-and-aggregate analysis identifies task-critical attention computation (Bick et al., 2025b); RADAR uses retrieval-sensitive heads to guide retention (Bick et al., 2026). RAD identifies redundant attention through self-speculative decoding; its self-distillation trains replacements with Transformer components frozen (Hoshino et al., 2025). Others use distillation loss or task validation (Li et al., 2026; Xia et al., 2026), or combine selection with training (Lan et al., 2026; Chen et al., 2026). We compare capability-selected layouts from a shared replacement bank at a fixed full-attention count, tracking task-specific recovery under a common repair protocol (Section 6.4). Recall/copying limitations (Arora et al., 2024a; Wen et al., 2025; Jelassi et al., 2024) and GenDistill's evaluation-dependent conclusions (Kostelec & Guo, 2026) motivate our targeted behavioral tests alongside aggregate scores.

3 Problem Setup: Sparse Transformer–SSM Hybrid Conversion

Hybrid architecture and attention layout. The teacher has L decoder layers F_i. The student retains attention at $\mathcal{A} = \{a_1 < \cdots < a_m\} \subseteq [L]$ and replaces the remaining attention modules with SSMs. Construction selects $\mathcal{A}$ under budget $|\mathcal{A}| = m$ and assembles candidate layers G_i from a locally trained *replacement bank*. Student layer $\widehat{F}_i^{\mathcal{A}}$ starts as F_i at retained positions and G_i elsewhere; repair updates its weights while keeping module types and positions fixed. The teacher remains fixed. We write $F_{a:b} = F_b \circ \cdots \circ F_a$, and analogously $\widehat{F}_{a:b}^{\mathcal{A}}$, for consecutive-layer composition. Equations use one-based indices; reported layouts are zero-based.

Layer outputs and supervision targets. The residual stream carries hidden representations between decoder layers, distinct from an SSM's recurrent state. Given input h, a teacher layer computes

$$y_i(h) = W_i^o m_i(N_i^{\text{in}}(h)), \qquad z_i(h) = h + y_i(h), \tag{1}$$

$$F_i(h) = z_i(h) + \text{MLP}_i\big(N_i^{\text{post}}(z_i(h))\big). \tag{2}$$

Here m_i is attention, $N_i^{\text{in}}, N_i^{\text{post}}$ are root-mean-square normalization (RMSNorm) (Zhang & Sennrich, 2019), and the *readout* W_i^o projects mixer features into the residual stream. The *attention write* y_i precedes residual addition; the *complete layer output* F_i includes both skip and MLP updates. We denote these teacher targets by $\bar{y}_i(x)$ and $\bar{h}_i(x)$ for token sequence x, with preceding layer input $\bar{h}_{i-1}(x)$. A replacement G_i substitutes an SSM mixer and may adapt its readout. A *layer boundary* is its complete output; a *block boundary* ends a consecutive training group. Group boundaries and retained attention positions need not coincide. Appendix A gives the full state recurrences.

4 Diagnosing Local Replacement and Cross-Layer Composition

4.1 Local Replacement: Readout Flexibility and Supervision Target

A replacement must supply useful outputs to the rest of its layer. Qwen preparation first matches individual attention-head outputs before the readout. From this common start, we vary readout

trainability and supervise either the projected mixer output (attention write, $\bar{y}_i$) or the complete layer output, $\bar{h}_i$. With surrounding components fixed, exact write matching implies full-output matching; approximate errors pass through the residual addition and MLP, so we measure both outputs.

The Local Qwen diagnosis study compares these choices at six random layers spanning shallow to deep positions, holding initialization, training duration, and probe batches fixed within each layer (Appendix D).

Adapting the readout improves both probes under either target (Table 1). At fixed readout trainability, full-layer versus write-only supervision improves full-output fitting and worsens write fitting in all six layers, with uneven magnitudes. Thus, better reproduction of the complete layer need not require closer imitation of its attention output. Readout adaptation and target choice have distinct effects on local fitting.

Table 1: Controlled readout $\times$ target experiment. Rows are training targets; columns probe the complete layer output (Full) or projected attention write (Attention) under each readout setting. Values average per-layer reductions in unsquared L2 loss (%) from the shared head-aligned starting point; higher is better.

	Frozen W_i^o		Trainable $\widetilde{W}_i^o$	
Training target	Full	Attention	Full	Attention
Attention write	+12.20	+11.81	+16.75	+16.61
Full-layer	+16.69	+9.23	**+20.27**	+13.16

To test the combined choices on downstream tasks, a 2,000-step experiment inserts six trained replacements into Qwen, retaining 22 attention layers. Adapting the readout and switching from write to full-layer supervision improves MMLU from 41.96 to 43.28 ($+1.32$ points, paired 95% CI $[0.91, 1.75]$) and HellaSwag normalized accuracy from 64.87 to 65.36 ($+0.49$, $[0.21, 0.78]$). WinoGrande and Lambada differences are inconclusive. This tests the combined repair; the factorial above separates its effects on local fitting (Appendix D).

An auxiliary single-seed comparison finds that dual-target supervision improves attention-write fitting at all six layers and complete layer output fitting at five over full-only training, without consistent task gains (Appendix D.3). We retain full-layer supervision with readout adaptation as the tested default: it targets the next layer's input and has combined task support.

4.2 Cross-Layer Composition: From Independent Fits to Student Computation

Local fitting supplies teacher residual-stream states, whereas the assembled hybrid passes on outputs produced by preceding student layers. Independent fits therefore leave the actual composed student computation unoptimized.

Exact layer-boundary agreement is sufficient, but not necessary, for agreement at a multi-layer output: one student layer can add an offset and the next remove it, preserving the teacher's final output. Appendix B gives a construction and an error bound for this property of composed functions.

Both blockwise and whole-model training optimize the student's composed computation. Section 6.2 tests the allocation choice: whether recovery benefits from training smaller groups first.

In an auxiliary pure-SSM Llama experiment, regional joint training from a common layerwise-distilled model raises MMLU from 34.1 to 37.9 in the middle span, versus 34.2 and 34.8 in early and late spans at the same training budget (Appendix E.3); other tasks favor different regions.

5 ReCompose: Diagnosis-Guided Repair

ReCompose prepares replacements before construction, then trains consecutive groups and the whole model at fixed attention positions (Figure 1). We describe Qwen here; Llama uses an existing Llamba bank and the subsequent stages with the trainable scopes in Appendix H.1.

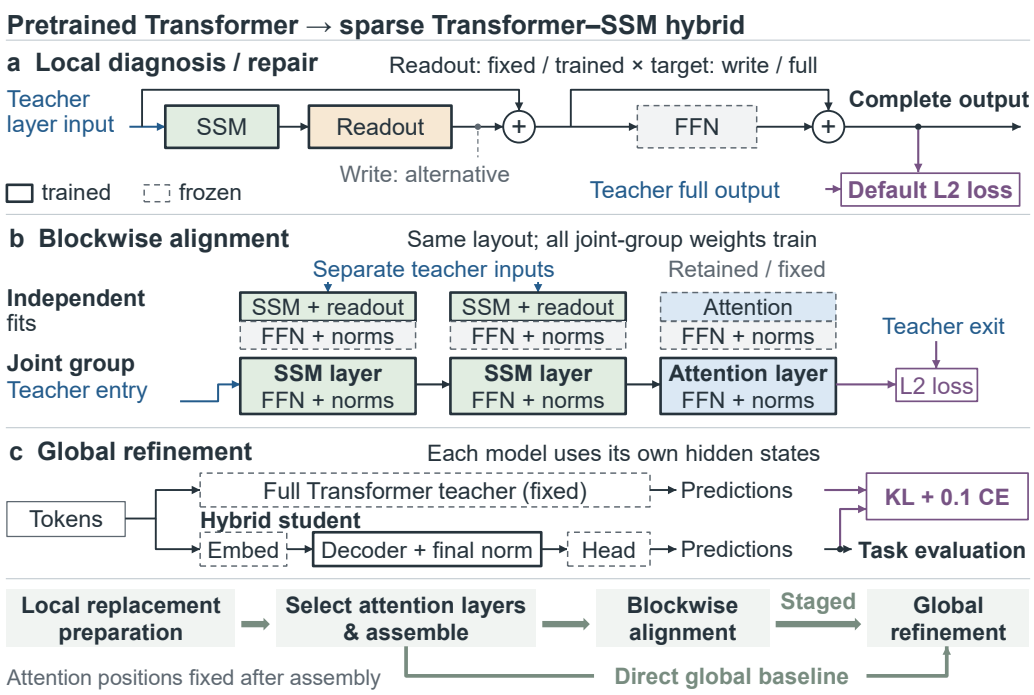


Figure 1: Qwen conversion and training paths. Both paths start from the same locally repaired assembly. Attention positions stay fixed; active block parameters, including retained attention, train jointly. Independent fits use teacher inputs; joint blocks propagate student states. The local default matches the complete layer output through fixed MLPs and norms (norms omitted). Global refinement uses each model's own states. KL + 0.1 CE is the Qwen training-path recipe; Llama uses KL alone and different trainable scopes (Appendix H.1).

5.1 Local repair: improve the replacement output

After operator and head-output alignment, our default trains the SSM and readout on teacher inputs against complete layer outputs:

$$\mathcal{L}_i^{\text{full}} = \mathbb{E}_x \, d\big(G_i(\bar{h}_{i-1}(x)), \bar{h}_i(x)\big). \tag{3}$$

Local and blockwise training use unsquared L2 distance averaged over tokens, $d(U, V) = T^{-1} \sum_{t=1}^{T} \|U_t - V_t\|_2$. Local MLP and normalization weights stay fixed; gradients pass through them to the SSM and readout. Appendix D distinguishes the local training budgets and settings.

5.2 Blockwise alignment: train composed layers

Our default partitions the student into contiguous blocks ending at retained attention positions, plus a suffix. Each span $[a, b]$ minimizes

$$\mathcal{L}_{a:b}^{\text{bdry}} = \mathbb{E}_x \, d\Big(\widehat{F}_{a:b}^{\mathcal{A}}(\bar{h}_{a-1}(x)), F_{a:b}(\bar{h}_{a-1}(x))\Big). \tag{4}$$

The teacher supplies the entrance and target; student layers process one another's outputs inside the block. All active block parameters train, including retained attention, MLPs, and norms. An attention-only single-layer block stays fixed; the suffix trains separately. Global refinement then trains interactions across blocks. Appendix E.2 gives partition controls.

5.3 Global refinement: train full-model behavior

Teacher and student process the same tokens using their own hidden states. With the teacher fixed, Qwen trains decoder layers and final normalization:

$$\mathcal{L}_{\text{global}} = \mathcal{L}_{\text{KL}} + \eta \mathcal{L}_{\text{CE}}. \tag{5}$$

Here *global* describes the full-model computation and prediction target; the embedding and prediction head remain fixed. KL matches teacher next-token distributions; CE is token cross-entropy with $\eta = 0.1$ for Qwen; Llama uses KL only (Appendix H.1).

Optional continued boundary supervision. During global refinement, we also test retaining supervision at selected block ends. This adds squared output errors normalized by initial-batch values, distinct from local/blockwise unsquared L2. We compare no penalty, a fixed coefficient, and a decreasing coefficient; all retain KL and CE. Appendices F and I specify the objectives.

6 Capability Recovery and Its Limits

6.1 Experimental protocol and study design

Four complementary studies connect conversion choices to recovery (Table 2). Qwen uses Qwen2.5-1.5B (Qwen et al., 2024); Llama uses Llama-3.1-8B-Instruct (Dubey et al., 2024) and provides qualitative cross-model support (Appendix A).

We evaluate multiple-choice accuracy with MMLU (Hendrycks et al., 2021), WinoGrande (Sakaguchi et al., 2020), and HellaSwag (normalized accuracy) (Zellers et al., 2019); completion with Lambada (Paperno et al., 2016); and long-context value retrieval with RULER needle-in-a-haystack (NIAH) tasks (Hsieh et al., 2024). MMLU uses micro accuracy unless explicitly identified as subject-macro. The shared-bank study tracks construction and recovery on full MMLU (14,042 examples) and 500 retrieval examples per length at 16k/32k. These evaluations participate in selection; fresh banks test retrieval beyond them.

Internal controls test training choices; teacher and published-model references describe capability levels (Section 2, Appendix G.2). Paired 95% intervals quantify evaluation uncertainty for the compared checkpoints, with definitions in Appendix A.2.

Table 2: Four complementary studies connect conversion choices to recovery.

Study	Starting point	Question and evidence
Local Qwen diagnosis	Common local initialization; task tests retain 22/28 attention layers	Readout × target; combined repair on six replacements; dual-target follow-up
Qwen training path	One locally repaired assembly; 4/28 attention layers	Same-start staged versus direct distillation; stage recovery
Qwen shared-bank layouts	Three selections from one bank, distinct from the training-path bank; 4/28 attention layers	Selection trade-offs, fixed-layout recovery, boundary objectives, fresh retrieval
Llama recovery	Existing Llamba bank; 4/32 attention layers; distinct trainable scope	Stage recovery, layer-selection controls, strict binding tests

6.2 Staged versus direct repair

From the same locally repaired Qwen assembly, six blockwise rounds followed by global refinement yield a 0.6–1.0-point MMLU advantage over direct global distillation at matched cumulative rounds 7–9 (Table 3). Both paths use the 1.5B teacher, FineWeb (Penedo et al., 2024), and the same global data windows, schedule, optimizer, prediction objective, and round-final evaluation (Appendix F.1).

With local preparation shared, this comparison tests the allocation of post-assembly training. Each round presents 327.68M tokens, counting the data window reused by block jobs once. Exposure is matched; computation differs because block jobs repeat teacher inference with different update scopes. The three columns are successive endpoints of one training trajectory per arm. Direct training retains higher Lambada accuracy at these endpoints, while the gap narrows: the staged path's MMLU gain is task-specific.

Table 3: Qwen round-final accuracy (%) at matched post-assembly exposure. Both paths start from the same locally repaired assembly; each column matches cumulative rounds, teacher and data source. Computation is not matched.

Task	Training path / cumulative round	7	8	9
MMLU	Direct whole-model	49.5	49.7	50.0
	ReCompose: blockwise then whole-model	50.5	50.3	50.7
Lambada	Direct whole-model	50.9	50.7	51.0
	ReCompose: blockwise then whole-model	46.8	48.7	49.8

6.3 Capability recovery across tasks and stages

With four retained attention layers, Qwen and Llama recover knowledge, commonsense, and completion accuracy (Table 4). Blockwise alignment improves MMLU but leaves large completion deficits; global refinement brings larger Lambada gains. Its MMLU effect differs: Qwen decreases from 51.7 to 50.7, whereas Llama improves from 60.2 to 60.5 from its actual blockwise starting checkpoint.

Table 4: Qwen2.5-1.5B and Llama-3.1-8B-Instruct recovery (accuracy, %). Headers show student attention budgets; teachers use full attention. Qwen follows blockwise round 6 to global round 3 (nine post-assembly rounds); Llama follows blockwise round 8 to global round 3 (eleven post-assembly rounds). HellaSwag is normalized accuracy. The dash denotes the Qwen round-final checkpoint's unreported normalized score; the supplementary score below uses a different saved checkpoint (Appendices G, H).

	Qwen (4/28 attention)				Llama (4/32 attention)			
Stage	MMLU	Hella.	Wino.	Lamb.	MMLU	Hella.	Wino.	Lamb.
Assembly	48.9	62.4	58.4	14.1	58.7	75.7	67.9	26.5
Blockwise	51.7	63.5	59.7	17.5	60.2	76.2	71.1	37.3
Global	50.7	–	62.9	49.8	60.5	77.0	70.9	65.2
Teacher	59.1	67.6	62.3	61.8	68.3	79.6	73.5	72.3
Qwen global round 3, separately saved checkpoint: HellaSwag normalized 64.6.								

Qwen evaluates the complete recipe; Llama tests blockwise and global recovery from an existing Llamba bank with a different trainable scope. Stage effects also vary across Qwen selections (Figure 2). Appendix G.2 compares the recovered capability levels with published models: relative rankings vary by task, and the source training and evaluation protocols differ.

6.4 Construction profiles and recovery

At four retained attention layers, selection shapes the initial capability trade-off. Qwen layouts from the same replacement bank favor MMLU, retrieval, or a balance of both (Table 5). MMLU selection has no measured retrieval success; the other selections exchange some MMLU accuracy for retrieval.

Table 5: Starting profiles from a shared Qwen replacement bank at four attention layers (accuracy, %). The displayed evaluations participate in selection; layout indices are zero-based.

Selection	Retained layers	MMLU	Retrieval 16k	32k
MMLU	$\{9, 17, 21, 22\}$	50.04	0.0	0.0
Retrieval	$\{6, 14, 19, 23\}$	32.53	100.0	81.0
Mixed	$\{6, 14, 19, 21\}$	44.00	96.4	72.0

The three assemblies retain this retrieval ordering on fresh 32k examples spanning different positions of the target information, with both repeated filler and essay text (Appendix I). Searches using MMLU, question answering, and perplexity also yield different knowledge and text-completion profiles on Qwen and Llama (Appendix C).

Llama controls further show capability-specific layer importance. With three attention positions fixed, layer 17 gives the best MMLU score among tested neighbors, consistent with prior sensitivity analysis (Bick et al., 2025b); none restores long-context retrieval. These construction-only controls use a limited MMLU screen. Selecting Llama layers for retrieval instead yields a different capability profile (Appendices H.2 and H.4).

We compare the three selected layouts under a common repair protocol, including shared block-partition and checkpoint-selection rules. Training changes their profiles without changing the retained attention positions (Figure 2). Blockwise alignment improves MMLU for all three layouts; global refinement partly restores retrieval even from the zero-scoring MMLU selection, reaching 23.2% at 16k and 8.8% at 32k. The other layouts attain stronger retrieval while preserving different MMLU levels. For the retrieval-selected layout, SQuAD completion declines during blockwise training and recovers during global refinement, illustrating non-monotonic task recovery (Appendix I.3). Fresh repaired-model tests follow in Section 6.5.

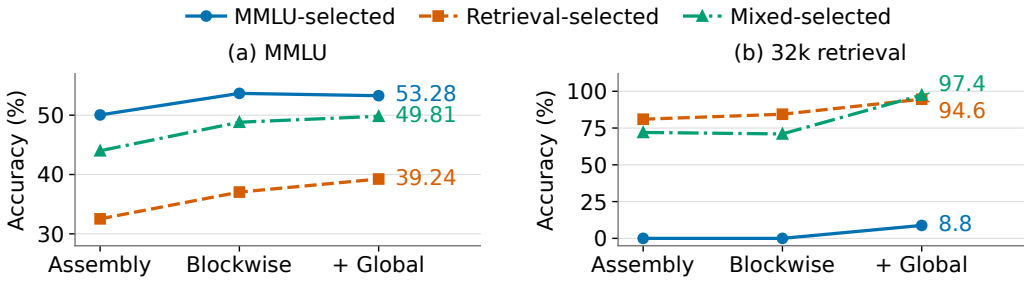


Figure 2: Qwen shared-bank study: capability profiles with each layout fixed during training. These evaluations participate in selection. Global refinement is unpenalized; stage spacing is not equal duration. The MMLU axis starts above zero.

Continued boundary supervision. In the Qwen training-path study, relative to unpenalized global refinement, a fixed boundary penalty improves MMLU by 0.3–0.8 points over six rounds while its Lambada deficit narrows from 1.1 to 0.1 (Appendix F.3). In the shared-bank study, the penalty improves retrieval-selected Qwen's 32k score by 2.0 points (paired 95% CI $[0.8, 3.4]$), but lowers retrieval for the MMLU and mixed selections. The same calibration procedure thus yields different outcomes across the tested layouts (Appendix I). Within the MMLU- and retrieval-selected models, many MMLU correctness disagreements between the two global objectives involve tied top option scores; they should not be read directly as changes in knowledge (Appendix I.5).

6.5 Fresh and demanding tests expose remaining failures

Aggregate gains can conceal persistent failures. Global refinement improves mixed-layout Qwen's value retrieval on fresh 32k-token contexts with repeated filler or essay text (Appendix I.4). Matching the target as a whole number, rather than part of a longer number, leaves all correctness labels unchanged; scoring only the first number changes one label but preserves the recovery pattern (Appendix I.5). All three mixed-layout checkpoints fail when the target appears in the essay's first quarter (Table 6).

Llama ranks one correct four-value answer against eight alternatives. Appendix J defines the scoring and interventions. Ordinary alternatives change all values, so recovering one association can suffice; hard alternatives change only one. From blockwise to global round 5, hard-task accuracy improves by 7.8 points ($[4.6, 11.0]$), but remains near random and far below the teacher, despite much stronger ordinary ranking.

When a queried value changes, correct choices in both original and changed contexts rise from 14/64 after blockwise training to 30/64 at global round 5, versus the teacher's 62/64 (Appendix J). This partial recovery illustrates why evidence changes, position, and association difficulty complement aggregate scores.

Table 6: Recovery boundaries (% except counts). Qwen: 256 fresh retrieval examples per format, not used for selection, including 64 in the earliest essay-text quarter. Llama: candidate ranking on 500 shared dossier examples; random accuracy is 11.1%.

Mixed-layout Qwen	Repeated filler	Essay text	Earliest essay quarter
Blockwise	67.19	53.91	0/64
+ Global, no penalty	96.88	64.06	0/64
+ Global, with penalty	96.88	66.80	0/64

Llama binding	Ordinary	Hard
Blockwise	25.2	4.0
+ Global, round 3	50.4	8.8
+ Global, round 5	58.2	11.8
Teacher	100.0	90.2

7 Practical Lessons and Limitations

Practical lessons. The local controls support readout adaptation and complete-layer supervision (Section 4). In the same-start Qwen comparison, early blockwise training yields higher MMLU, while direct global distillation reaches higher Lambada accuracy at the tested endpoints (Section 6.2). These choices use the existing parameters and attention budget.

The shared design idea is to match what a layer or group passes onward and choose which components adapt together. The dual-target and boundary comparisons show how additional output constraints change local fitting and downstream behavior. Evaluate those constraints against the intended tasks.

Choose layouts and checkpoints for the intended capabilities. Evaluate construction before and after training: initial retrieval failure can be partly repaired without changing selected layers (Section 6.4), and completion-favored checkpoints need not maximize MMLU. Validate recovery on fresh examples, varied evidence positions, alternatives with one incorrect binding, and contexts whose correct answer changes.

Limitations. Each training arm has one trajectory; reported intervals cover checkpoint-conditional evaluation uncertainty, leaving training-seed variation unmeasured. Shared-bank runs compare complete layout configurations, without isolating individual-layer contributions or establishing search-procedure superiority under matched budgets. Fresh repaired-model retrieval covers only mixed Qwen. The contribution of learned compensation and retained attention to the observed recovery remains unresolved. Final-model deployment efficiency and superiority over complete external pipelines remain unmeasured.

8 Conclusion

We diagnose capability loss during conversion and translate the findings into ReCompose. Readout adaptation improves local fitting; the combined readout and target repair improves downstream accuracy. ReCompose trains consecutive complete layers before global distillation, supported by the same-start Qwen comparison. Qwen and Llama recover capabilities with four attention layers. Attention-layer selection shapes initial capability trade-offs, and training can recover some losses without changing the selected layers. These findings guide supervision, training allocation and model selection within the observed positional and binding limits.

AI USE STATEMENT

Generative AI assisted theoretical derivations and proofs, methodological feedback, code development, result interpretation, literature review, and manuscript and figure preparation. The authors led the research, reviewed the AI-assisted work, and take responsibility for the final content.

ETHICS STATEMENT

This study involves no human subjects or private personal data.

REPRODUCIBILITY STATEMENT

Section 5 specifies the training objectives and trainable components, and Section 6 identifies the study settings. Appendix A describes implementation and evaluation conventions, including statistical uncertainty. Appendix B gives the assumptions and proof for the composition analysis. The subsequent experimental appendices report selection procedures, training settings, evaluation protocols, and additional results, including the scope of the comparisons and the limits of repeated-run evidence.

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 *Proceedings of 12th International Conference on Learning Representations (ICLR)*. ICLR, 2024a. URL https://arxiv.org/abs/2312.04927.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1763–1840. PMLR, 2024b. URL https://proceedings.mlr.press/v235/arora24a.html.

Parsa Ashrafi Fashi, Utkarsh Saxena, Mehdi Rezagholizadeh, Aref Jafari, Akash Haridas, Mingyu Yang, Vansh Bhatia, Guihong Li, Vikram Appia, and Emad Barsoum. Long-context aware upcycling: A new frontier for hybrid LLM scaling. *arXiv preprint arXiv:2604.24715*, 2026. URL https://arxiv.org/abs/2604.24715v1.

Akhiad Bercovich, Tomer Ronen, Talor Abramovich, Nir Ailon, Nave Assaf, Mohammed Dabbah, Ido Galil, Amnon Geifman, Yonatan Geifman, Izhak Golan, Netanel Haber, Ehud Dov Karpas, Roi Koren, Itay Levy, Pavlo Molchanov, Shahar Mor, Zach Moshe, Najeeb Nabwani, Omri Puny, Ran Rubin, Itamar Schen, Ido Shahaf, Oren Tropp, Omer Ullman Argov, Ran Zilberstein, and Ran El-Yaniv. Puzzle: Distillation-based NAS for inference-optimized LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3806–3830. PMLR, 2025. URL https://proceedings.mlr.press/v267/bercovich25a.html.

Aviv Bick, Kevin Y. Li, Eric P. Xing, J. Zico Kolter, and Albert Gu. Transformers to SSMs: Distilling quadratic knowledge to subquadratic models. *Advances in Neural Information Processing Systems*, 37:31788–31812, 2024. URL https://arxiv.org/abs/2408.10189v2.

Aviv Bick, Tobias Katsch, Nimit Sharad Sohoni, Arjun D Desai, and Albert Gu. Llamba: Scaling distilled recurrent models for efficient language processing. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025a. URL https://arxiv.org/abs/2502.14458v1.

Aviv Bick, Eric Xing, and Albert Gu. Understanding the skill gap in recurrent language models: The role of the gather-and-aggregate mechanism. In *Forty-second International Conference on Machine Learning*, pp. 4324–4344, 2025b. URL https://proceedings.mlr.press/v267/bick25a.html.

Aviv Bick, Eric P. Xing, and Albert Gu. Retrieval-aware distillation for Transformer–SSM hybrids. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.11374v1.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Yingfa Chen, Zhen Leng Thai, Zihan Zhou, Zhu Zhang, Xingyu Shen, Shuo Wang, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Hybrid linear attention done right: Efficient distillation and effective architectures for extremely long contexts. *arXiv preprint arXiv:2601.22156*, 2026. URL https://arxiv.org/abs/2601.22156v1.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/98ed250b203d1ac6b24bbcf263e3d4a7-Abstract-Conference.html.

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

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A hybrid-head architecture for small language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f32def07618040e540e0a6182e290562-Abstract-Conference.html.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783v1.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation. Zenodo, 2023. URL https://zenodo.org/records/10256836.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, Anna Golubeva, Vasudev Shyam, James Whittington, Jonathan Pilault, and Beren Millidge. The Zamba2 suite: Technical report. *arXiv preprint arXiv:2411.15242*, 2024. URL https://arxiv.org/abs/2411.15242.

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

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

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Yuxian Gu, Qinghao Hu, Haocheng Xi, Junyu Chen, Shang Yang, Song Han, and Han Cai. Jet-nemotron: Efficient language model with post neural architecture search. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=WZQXaTNYEB.

Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: A unified solution for structured web data extraction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011. URL https://www.microsoft.com/en-us/research/publication/from-one-tree-to-a-forest-a-uni%EF%AC%81ed-solution-for-structured-web-data-extraction/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300v3.

Yuichiro Hoshino, Hideyuki Tachibana, Muneyoshi Inahara, and Hiroto Takegawa. RAD: Redundancy-aware distillation for hybrid models via self-speculative decoding. *arXiv preprint arXiv:2505.22135*, 2025. URL https://arxiv.org/abs/2505.22135v1.

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*, 2024. URL https://arxiv.org/abs/2404.06654v3.

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*, pp. 21502–21521. PMLR, 2024. URL https://proceedings.mlr.press/v235/jelassi24a.html.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, 2020. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372/.

Juan Gabriel Kostelec and Qinghai Guo. When perplexity lies: Generation-focused distillation of hybrid sequence models. *arXiv preprint arXiv:2603.26556*, 2026. URL https://arxiv.org/abs/2603.26556v2.

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

Barak Lenz, Opher Lieber, Alan Arazi, et al. Jamba: Hybrid transformer–Mamba language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a9ed43fa31dc8b4a7d7a673d713dcb5f-Abstract-Conference.html.

Yanhong Li, Songlin Yang, Shawn Tan, Mayank Mishra, Rameswar Panda, Jiawei Zhou, and Yoon Kim. Distilling to hybrid attention models via KL-guided layer selection. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b08940ce0f8f2bd499f2e5440eb38ac5-Abstract-Conference.html.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2102.05426v2.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

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 (volume 1: Long papers)*, pp. 1525–1534, 2016. URL https://aclanthology.org/P16-1144/.

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. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024. URL https://arxiv.org/abs/2406.17557.

Qwen, An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115v1.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

SLAM Labs, Oleksiy Ostapenko, Raymond Li, Torsten Scholak, Alireza Mousavi-Hosseini, Aman Tiwari, Denis Kocetkov, Joel Lamy Poirier, Kelechi Ogueji, Nanda H Krishna, Rafael Pardinas, Sathwik Tejaswi Madhusudhan, Shruthan Radhakrishna, Srinivas Sunkara, and Valérie Bécaert. Super apriel: One checkpoint, many speeds. *arXiv preprint arXiv:2604.19877*, 2026. URL https://arxiv.org/abs/2604.19877v1.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. URL https://arxiv.org/abs/1706.03762.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. The Mamba in the llama: Distilling and accelerating hybrid models. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/723933067ad315269b620bc0d2c05cba-Paper-Conference.pdf.

Kaiyue Wen, Xingyu Dang, and Kaifeng Lyu. RNNs are not transformers (yet): The key bottleneck on in-context retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=h3wbI8Uk1Z.

Xiaojie Xia, Huigang Zhang, Chaoliang Zhong, Jun Sun, and Yusuke Oishi. Distill-then-replace: Efficient task-specific hybrid attention model construction. *arXiv preprint arXiv:2601.11667*, 2026. URL https://arxiv.org/abs/2601.11667.

Mingyu Yang, Mehdi Rezagholizadeh, Guihong Li, Vikram Appia, and Emad Barsoum. Zebra-Llama: Towards extremely efficient hybrid models. *arXiv preprint arXiv:2505.17272*, 2025. URL https://arxiv.org/abs/2505.17272v2.

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

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1910.07467.

Michael Zhang, Simran Arora, Rahul Chalamala, Alan Wu, Benjamin Spector, Aaryan Singhal, Krithik Ramesh, and Christopher Ré. LoLCATs: On low-rank linearizing of large language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10254v3.

Zhongzhu Zhou, Qingyang Wu, Junxiong Wang, Mayank Mishra, Shuaiwen Leon Song, Ben Athiwaratkun, and Chenfeng Xu. Taylor-Calibrate: Principled initialization for hybrid linear attention distillation. *arXiv preprint arXiv:2606.16429*, 2026. URL https://arxiv.org/abs/2606.16429v1.

A Notation and Evaluation

The appendices provide notation and evaluation conventions, followed by construction, local, blockwise and global controls. Qwen and Llama recovery details precede the shared-bank layout study, strict binding tests and runtime measurements. Settings are given alongside the comparisons they define.

A.1 Notation and study identities

Assembly and state recurrences. Writing $\mathcal{R} = [L] \setminus \mathcal{A}$, assembly initializes

$$\widehat{F}_i^{\mathcal{A}} = \begin{cases} F_i, & i \in \mathcal{A}, \\ G_i, & i \in \mathcal{R}, \end{cases} \qquad \widehat{F}_{1:L}^{\mathcal{A}} = \widehat{F}_L^{\mathcal{A}} \circ \cdots \circ \widehat{F}_1^{\mathcal{A}}. \tag{6}$$

Subsequent repair updates student parameters, including retained layers, while F_i remains the teacher function. For token sequence x, $\bar{h}_0(x) = \mathrm{Emb}(x)$, $\bar{h}_i(x) = F_i(\bar{h}_{i-1}(x))$, and $\bar{y}_i(x) = y_i(\bar{h}_{i-1}(x))$. Independent preparation supplies $\bar{h}_{i-1}$ to each replacement; the assembled student instead computes $\hat{h}_i = \widehat{F}_i^{\mathcal{A}}(\hat{h}_{i-1})$.

Indexing and experimental comparisons. Equations use one-based layers $i \in [L]$; reported layouts use zero-based indices. The Qwen training-path study uses $\mathcal{A} = \{0, 9, 17, 21\}$ in a 28-layer Qwen2.5-1.5B model. The Llama recovery study uses Llama-3.1-8B-Instruct with $\{9, 11, 13, 17\}$ and a reused Llamba replacement bank. Its teacher, replacement source, local protocol and scale differ from Qwen's. The Qwen shared-bank layout study uses a different replacement bank, shared across its MMLU-, retrieval-, and mixed-selected layouts (Appendix I).

A replacement bank collects independently prepared SSM replacements. The local factorial and dual-target follow-up use separate 1,000-step checkpoints; the two-arm splice uses 2,000-step checkpoints and retains 22 attention layers. The training-path bank uses longer local preparation (Appendix D.4). The two Qwen banks use different checkpoints of that preparation.

A.2 Evaluation and statistical conventions

The standard-task suite uses the LM Evaluation Harness (Gao et al., 2023) and reports accuracy on full MMLU, Lambada and WinoGrande, plus length-normalized accuracy on HellaSwag. Unless stated otherwise, accuracies are percentages and their differences are percentage points. Search-time MMLU subsets are never reported as full MMLU. MMLU scores the letter continuations A–D by summing their token log-likelihoods; exact ties select the first maximum in that order. Appendix I.5 examines saved scores under this rule. Table headings abbreviate Lambada, WinoGrande and HellaSwag as Lamb., Wino. and Hella., respectively. $\mathrm{PPL}_{\mathrm{tok}}$ measures token-level perplexity on Lambada text: the first 128 test examples are tokenized and truncated to 256 tokens, token NLL is summed over evaluated positions, and we report the exponential of mean token negative log-likelihood. This differs from the Harness Lambada continuation PPL, which scores only the final word, and from training-domain or WikiText perplexity.

Statistical inference. MMLU micro accuracy weights examples equally; subject-macro accuracy weights the 57 subjects equally. Within-subject bootstrap resampling preserves this distinction. The local splice and Qwen training-path boundary comparisons use 10,000 paired resamples; the splice uses seed 42. The MMLU- and NIAH-selected Qwen endpoint comparisons use 20,000 resamples and preserve all 14,042 MMLU occurrences. Fresh-sample retrieval comparisons also use 20,000; selection-profile intervals use 10,000. Binding tests use 10,000 original-example resamples, as specified in Appendix J.

All intervals condition on the compared checkpoints and quantify evaluation sampling, not training-run variation. Rounds along a trajectory are correlated. Selection-used evaluations characterize the selected models; fresh repaired-model retrieval tests cover the mixed layout. Task-specific scoring checks accompany the relevant results in Appendix I.5.

A.3 Replacement implementation and training scope

Qwen training inputs and scope. Table 7 summarizes the three training stages.

Table 7: Inputs, targets, and trainable scope of the Qwen repair protocol. The teacher stays frozen. Here $\bar{h}_j$ denotes the complete decoder output after residual and MLP updates. Blockwise training propagates student states from the teacher block input. Local controls freeze the readout or substitute an attention-write target (Table 1).

Training step	Student input	Teacher target	Trainable parameters
Local repair	Layer input $\bar{h}_{i-1}$	Layer output $\bar{h}_i$	SSM mixer and W_i^o
Blockwise alignment	Block input $\bar{h}_{a-1}$	Block output $\bar{h}_b$	All active-block parameters
Logit distillation	Token sequence x	Next-token distribution	Decoder layers and final norm

Replacement implementation and initialization. The Qwen implementation uses a layerwise Mamba-2 (Dao & Gu, 2024) wrapper. Its model width matches the teacher's hidden size, and its query/key and value head counts equal the teacher's attention-head count. The value-head width is the model width divided by that head count. Its constructor uses expansion one, identity activation, and chunk size 128. The training configuration uses a state dimension of 64 and a convolution width of 4.

Each replacement copies its teacher layer's output projection as an independently trainable student readout; the teacher copy stays frozen. The adapter applies this readout to the mixer's pre-projection features, bypassing the internal projection. It does not call the teacher's query/key/value projections or rotary embeddings in this layerwise path. The copied decoder retains the teacher MLP and norms, frozen during local repair and trainable in active Qwen blocks. Operator/head-output warm starts precede the local target choices (Appendix D).

Parameter counts. The counts below include all registered student parameters, with the tied embedding and language-model head counted once. The internal projection remains registered but is bypassed by the external readout described above.

Model	Registered parameters	Bypassed projection parameters
Qwen teacher	1,543,714,304	0
Six-replacement student	1,581,662,792	14,155,776
24-replacement student	1,695,508,256	56,623,104

Thus the sparse student registers approximately 1.70B parameters, of which 1.64B participate in the external-readout computation. Qwen2.5-1.5B denotes its teacher source, not the converted model's parameter count.

The Llama replacement bank uses the donor Llamba mixer configuration and its own output projection. Appendix H describes its initialization and training conditions.

B Layerwise and Blockwise Matching

Matching every intermediate layer is a stronger constraint than matching the transformation of a whole block. The following bound and counterexample make this distinction precise: local discrepancies can propagate, but need not survive at the block output. This provides a rationale for testing multi-layer training.

Proposition 1 (Layerwise matching is sufficient, not necessary)**.** *Consider a block $[a, b]$ under a fixed attention layout $\mathcal{A}$. For input x, let*

$$\epsilon_i(x) = \left\| \widehat{F}_i^{\mathcal{A}}(\bar{h}_{i-1}(x)) - F_i(\bar{h}_{i-1}(x)) \right\|_2, \tag{7}$$

which is zero for unchanged retained attention layers at construction. If alignment updates a retained layer, its discrepancy is included in the same bound. Suppose each $\widehat{F}_i^{\mathcal{A}}$ is K_i-Lipschitz on the local region containing teacher-generated and hybrid residual-stream states. Then

$$\left\| \widehat{F}_{a:b}^{\mathcal{A}}(\bar{h}_{a-1}) - F_{a:b}(\bar{h}_{a-1}) \right\|_2 \le \sum_{i=a}^{b} \left(\prod_{j=i+1}^{b} K_j \right) \epsilon_i(x). \tag{8}$$

Exact one-layer boundary matching on all layers implies exact block matching, but exact block matching does not in general imply exact one-layer matching.

Proof. Write $\bar{h}_i = F_{a:i}(\bar{h}_{a-1})$ and $\hat{h}_i = \widehat{F}^{\mathcal{A}}_{a:i}(\bar{h}_{a-1})$, with shared input $\bar{h}_{a-1} = \hat{h}_{a-1}$, and let $e_i = \|\hat{h}_i - \bar{h}_i\|_2$. Adding and subtracting $\widehat{F}^{\mathcal{A}}_i(\bar{h}_{i-1})$ gives

$$e_i \le \left\| \widehat{F}^{\mathcal{A}}_i(\hat{h}_{i-1}) - \widehat{F}^{\mathcal{A}}_i(\bar{h}_{i-1}) \right\|_2 + \epsilon_i \le K_i e_{i-1} + \epsilon_i. \tag{9}$$

Unrolling the recurrence proves the bound. If every $\epsilon_i = 0$, then $e_b = 0$.

For the converse, take $z = (z_1, z_2)$, $\phi(t) = \tanh(t)$, and $c \neq 0$:

$$F_a(z) = \begin{bmatrix} z_1 \\ \phi(z_2) + c \end{bmatrix}, \qquad F_b(u) = \begin{bmatrix} u_1 + u_2 \\ u_2 \end{bmatrix}.$$

Let the first hybrid layer omit the offset, $\widehat{F}_a(z) = [z_1, \phi(z_2)]^\top$, and let the second compensate,

$$\widehat{F}_b(u) = \begin{bmatrix} u_1 + u_2 + c \\ u_2 + c \end{bmatrix}.$$

Then $\|\widehat{F}_a(z) - F_a(z)\|_2 = |c|$, while $\widehat{F}_b(\widehat{F}_a(z)) = F_b(F_a(z))$. A later map can therefore compensate for an earlier layer's discrepancy at the block boundary. The nonzero local error is compatible with an exact two-layer composition, which explains why a block objective can admit solutions excluded by exact layerwise matching. This is an illustrative construction, not a model of an identified compensation mechanism in the trained hybrid.

C Selection Objective and Initial Capabilities

Selections made with different objectives have different starting profiles. In the construction controls accompanying the Qwen training-path study and Llama recovery study, MMLU selection gives the highest MMLU score, whereas perplexity selection gives the lowest perplexity and highest Lambada accuracy (Table 8). These starting profiles motivate evaluating recovery across the target tasks.

These searches evaluate complete layer sets under an attention-count upper bound, using measured proxy performance as fitness. The Qwen MMLU search uses a subset of MMLU; Table 8 reports full evaluations. The Qwen SQuAD-selected and Llama PPL-selected models retain three attention layers, and the others retain four. All rows precede blockwise and global training; the Qwen MMLU assembly already includes local repair. The SQuAD rows use the completion-formatted proxy of Arora et al. (2024b), derived from SQuAD (Rajpurkar et al., 2016), and score whether generated text contains the answer.

Table 8: Selected layouts before blockwise and global training. Three-attention rows were allowed under the at-most-four search budget; attention counts therefore differ across some comparisons. Accuracies are percentages; HellaSwag uses length-normalized accuracy. $\mathrm{PPL}_{\mathrm{tok}}$ follows Appendix A.

Family	Proxy / retained layers	MMLU	Lamb.	Wino.	Hella.	PPL
Qwen	MMLU $\{0, 9, 17, 21\}$	48.9	14.1	58.4	62.4	31.28
Qwen	SQuAD $\{14, 15, 23\}$	32.6	22.0	56.8	59.1	37.09
Qwen	PPL $\{0, 14, 19, 27\}$	33.2	27.5	58.8	61.3	29.86
Llama	MMLU $\{9, 11, 13, 17\}$	58.7	26.5	67.9	75.7	35.25
Llama	SQuAD $\{1, 4, 15, 17\}$	52.2	28.0	67.1	75.8	63.12
Llama	PPL $\{13, 15, 30\}$	38.7	33.4	65.0	75.4	29.61

The Qwen training-path study and Llama recovery study use Qwen $\{0, 9, 17, 21\}$ and Llama $\{9, 11, 13, 17\}$. The Llama controls in Appendix H examine sensitivity to the retained layer indices. The Qwen shared-bank layout study fixes the count at four and evaluates MMLU-, NIAH-, and mixed-selected assemblies from a shared, different replacement bank (Table 5). Appendix I gives those searches and their subsequent training.

D SEPARATING THE LOCAL TRAINING CHOICES

The local factorial separates readout adaptation from supervision target. A two-arm splice tests their combined task benefit; a dual-target follow-up asks whether better local matching selects the better assembled model.

D.1 WHAT THE LOCAL CONTROLS ISOLATE

Three supervision locations. Head, mixer, and layer outputs refer to different points in the same decoder computation:

Location	Teacher target	Role in the local study
Head	Each head's attention-weighted value vector, before output projection	Common initialization before the target comparison
Mixer	Combined attention output after projection, before residual addition	Attention-write target and probe
Layer	Complete output after residual additions and the MLP	Full-layer target and probe

Head matching compares the SSM's gated pre-projection features, split into heads, with teacher head outputs using unsquared L2 averaged over heads and tokens. The SSM replaces the whole attention module. Head matching supplies the common initialization for the subsequent target comparison.

The 2×2 comparison in Table 1 uses zero-based layers $\{1, 8, 18, 21, 22, 27\}$. We randomly selected six layers spanning shallow, middle, and deep positions, with the sample size limited by the available training time. The same sampled set is used for all local comparisons. For each layer, all branches share the operator warm start (50 steps) and head-output alignment (4,000 steps). All four target/readout branches then train for 1,000 steps on the shared per-layer data stream. Only the SSM mixer and, in the adapted branches, the output projection are trainable; the MLP and both RMSNorms remain fixed. The factorial branches use length-2,048 sequences, batch size one and eight accumulation steps. The mixer learning rate starts at 5×10^{-5}, with 5% warmup and cosine decay to 10^{-5}; the readout rate is 0.3 times the mixer rate. Each target branch starts with a fresh optimizer.

The longer Qwen replacement-bank preparation used 50 operator, 6,000 head-output, and 1,000 attention-write steps before readout adaptation and full-layer continuation. The controlled comparison uses the shared 4,000-step head preparation above and varies subsequent readout/target choices.

The frozen- and adapted-readout optimizer sets contain 9,472,524 and 11,831,820 parameters per replacement, respectively (+24.9%). Both include the 2,359,296-parameter internal projection bypassed by the inherited readout. Excluding this unused projection, 7,113,228 and 9,472,524 parameters participate in the local loss (+33.2%). Unfreezing the inherited 1536×1536 readout exposes 2,359,296 existing parameters per layer to training. Across 24 replacements, these total 56.6M, or 3.34% of the 1.696B registered student parameters; unfreezing adds no inference-time parameters. The downstream comparison below tests the combined change in readout trainability and target.

The two teacher targets are the projected attention write before residual addition and the complete decoder output after residual and MLP updates. Both training and cross-probing use

$$d(U, V) = \frac{1}{T} \sum_{t=1}^{T} \|U_t - V_t\|_2, \tag{10}$$

where T counts token positions across the minibatch. The loss takes an L2 norm along the hidden dimension and then averages.

All branches and their common starting point are cross-probed on 96 fixed batches: 32 at each stream offset 0, 10,000 and 50,000. Table 1 averages per-layer relative loss reductions; every branch improves both means over the head-aligned start. This measures further fitting, without a matched continuation of head supervision. We evaluate final checkpoints here and probe-best checkpoints in the longer local preparation below.

Table 9 separates the effects of readout adaptation and target choice. At all six layers, readout adaptation improves both probes under either target; changing to the full-layer target improves full-layer fitting and worsens attention-write fitting under either readout setting. Removing any one layer preserves the mean direction, but the effect sizes are uneven. From the three-decimal reported losses, changing only the target reduces full-layer loss by approximately 5.23% with a frozen readout and 4.43% with an adapted readout; excluding layer 1 gives approximately 1.67% and 0.86%, respectively. These are descriptive sensitivity calculations on the reported means.

Table 9: Pairwise effects in the local factorial comparison. Positive values reduce probe loss, averaged over six layers. The last row compares adapted-readout/ full-layer training with frozen-readout/attention-write training.

Changed training choice	Attention-write probe	Full-layer probe
Adapt readout, attention target fixed	+5.56%	+5.30%
Full-layer target, readout frozen	−2.99%	+5.23%
Adapt readout, full-layer target fixed	+4.39%	+4.50%
Adapt readout and use full-layer target	+1.57%	+9.41%

D.2 Do the combined choices improve task performance?

The Local Qwen diagnosis study includes a separate two-arm splice on zero-based layers $\{1, 8, 18, 21, 22, 27\}$. Each branch trains for 2,000 steps after the shared 50+4,000-step warm start; the factorial above uses 1,000-step checkpoints. Both branches train the SSM mixer. Full also trains the readout and targets the complete layer output; Rigid freezes the readout and targets the attention write. The experiment identifies their joint effect, not the individual contribution of readout adaptation or target selection. The combined repair improves MMLU and HellaSwag in the reported paired evaluation; WinoGrande and Lambada remain inconclusive (Table 10).

Table 10: Paired evaluation of the six-layer splice. Intervals use 10,000 bootstrap resamples and seed 42. They are unadjusted and condition on one trained checkpoint pair. Differences are Full minus Rigid in percentage points; subject-macro MMLU weights the 57 subjects equally, while the other rows report micro accuracy.

Task / aggregation	N	Full	Rigid	Δ	95% CI
MMLU micro	14,042	43.28	41.96	+1.32	[0.91, 1.75]
MMLU subject-macro	57 subj.	44.43	42.88	+1.55	[1.05, 2.05]
HellaSwag norm	10,000	65.36	64.87	+0.49	[0.21, 0.78]
WinoGrande	1,267	63.54	62.83	+0.71	[−1.03, 2.53]
Lambada	5,153	32.47	32.87	−0.41	[−0.87, 0.06]

Prediction quality beyond task accuracy. A separate fixed evaluation of the same two repair configurations measures perplexity and teacher–student probe KL (Table 11). Full lowers both perplexity and teacher–student probe KL, complementing the paired task comparison.

Table 11: Separate fixed evaluation of the six-replacement splice. $\mathrm{PPL}_{\mathrm{tok}}$ uses Appendix A's token-level definition; lower PPL and probe KL are better. These point estimates are separate from the evaluations used for the paired intervals in Table 10.

Configuration	$\mathrm{PPL}_{\mathrm{tok}}$	Probe KL
Rigid	28.55	0.1687
Full	27.56	0.1342

D.3 Dual-Target Supervision of Attention Writes and Complete Layer Outputs

Can supervising both outputs improve both local interfaces, and does this select a better assembled model? We compare three objectives with trainable readouts: write-only, full-only, and equal-weight

dual-target supervision (Dual),

$$\mathcal{L}_{\text{dual}} = \tfrac{1}{2}\mathcal{L}_{\text{write}} + \tfrac{1}{2}\mathcal{L}_{\text{full}}. \tag{11}$$

Each component is token-averaged unsquared L2 distance at its respective output.

Setup. This follow-up uses Qwen2.5-1.5B and zero-based layers $\{1, 8, 18, 21, 22, 27\}$, with a common warm start within each layer. All three arms train the mixer and readout for 1,000 optimizer steps with one fixed seed; MLPs and norms remain frozen. Task differences are point estimates without paired intervals. We splice the six replacements into the model, retaining attention at the other 22 layers. Dual-target versus full-only supervision is the primary target comparison. These checkpoints are separate from the readout-by-target factorial and the earlier 2,000-step Full/Rigid splice. Both losses supervise one replacement on teacher inputs.

Local agreement. Table 12 compares dual-target with full-only supervision on fixed probe batches, which are not disjoint validation. Dual-target supervision lowers write error in all six layers and full-layer error in five. Its write error remains higher than with write-only supervision in all six. Thus dual-target supervision can improve both probes relative to full-only training; task performance provides a separate test.

Table 12: Dual-target versus full-only supervision: relative error reductions $100(e_{\text{full}} - e_{\text{dual}})/e_{\text{full}}$, where e is probe error under the indicated training objective. Positive values mean lower error. The mean averages six per-layer relative reductions.

Layer	Write-error reduction	Full-layer-error reduction
1	+4.95%	+2.98%
8	+2.63%	+0.50%
18	+2.36%	+0.51%
21	+1.34%	+0.20%
22	+1.69%	+0.32%
27	+2.15%	−0.24%
Mean	+2.52%	+0.71%

Assembled behavior. Dual-target supervision scores below full-only on MMLU, HellaSwag and WinoGrande, and above it on Lambada (Table 13). The improvement in local fitting therefore accompanies mixed downstream changes.

Table 13: Single-seed six-replacement task comparison. Accuracies are percentages; Dual minus full-only differences are in percentage points, computed before rounding. MMLU micro is the manuscript's primary aggregation; subject-macro is supplementary here.

Metric	Write-only	Full-only	Dual	Dual − full-only
MMLU micro	42.36	42.86	42.52	−0.33
MMLU subject-macro	43.43	43.94	43.59	−0.36
HellaSwag normalized	64.98	65.10	64.94	−0.16
Lambada	32.70	31.75	32.19	+0.45
WinoGrande	63.61	64.17	62.90	−1.26

Full-only remains the tested ReCompose default.

D.4 Longer local training used for assembly

Both Qwen replacement banks use longer preparation than the controlled diagnostic in Appendix D.1. They use the 50/6,000/1,000-step operator/head/write initialization described there, Qwen2.5-1.5B as teacher, and length-2,048 FineWeb-Edu sequences. After initialization, each layer trains its SSM and readout on teacher inputs, with its MLP and normalizations fixed. Continuations load the preceding final weights and restart AdamW (weight decay 0.01).

Readout adaptation uses successive 2,000-, 4,000- and 1,000-update runs at effective batch eight. The mixer rate follows the 5×10^{-5} to 10^{-5} cosine schedule with 5% warmup, but the readout multiplier is 0.1. Full-layer supervision raises that multiplier to 0.3. Its initial 27,000 updates retain the same batch and mixer-rate range: one 1,000-update run, four 2,000-update runs and six 3,000-update runs; the last two use 1% warmup. Subsequent rounds each use 5,000 updates, batch eight with four accumulation steps (effective batch 32), and a mixer rate of 10^{-4} decaying to 2×10^{-5} after 5% warmup. The shared-bank layout study uses the bank after three such rounds; the training-path study uses three further rounds of the same preparation.

Assembly selects each layer's lowest mean full-layer L2 probe loss within its bank's last round, among scheduled probe checks and the final state. The fixed probe has 16 batches at each stream offset 0, 10,000 and 50,000. These batches come from the training stream; downstream task accuracy does not select the per-layer checkpoints. For the training-path bank, reconstructed token presentations are approximately 14.44M through frozen-readout alignment, 114.69M for readout adaptation including its continuations, and 2.408B for full-layer training, or 2.538B per replacement layer. FineWeb stream resets make this repeated exposure rather than globally unique data.

E Blockwise Alignment and Partition Controls

Blockwise alignment changes both the supervised output and the span trained together: several student layers must produce the teacher's block-exit state. The controls below compare partitions whose boundaries do or do not coincide with retained attention layers. A pure-SSM experiment then tests whether joint training recovers capabilities beyond a layerwise-distilled starting point, and whether recovery is greater near a predicted critical region.

E.1 Training the composed student block

The retained attention layout is $\{0, 9, 17, 21\}$ in zero-based indexing. The active blocks are $[1, 9]$, $[10, 17]$, $[18, 21]$, and the tail $[22, 27]$; indices are zero-based and inclusive. Attention-only layer 0 remains fixed as a singleton in the model. One block is active per training job. The teacher runs without gradients to supply the block's entry and exit states. Within the student block, each layer receives the preceding student layer's output.

The Qwen training-path study trainer computes the difference between student and teacher block outputs in float32, takes the unsquared L2 norm over the hidden dimension, and averages over batch and token positions. This is the distance d in Eq. 4; its probe loss uses the same reduction. It is distinct from the normalized squared-error boundary penalty used in logit distillation. Llama uses the same block-output distance but a different trainable scope (Table 20).

The Qwen training-path study runs train all parameters of the active block, including replacement mixers, output projections, MLPs, norms, and retained attention. The learning rate ranges from 3×10^{-5} to 5×10^{-6}, with 3% warmup and a 30% stable phase before decay. Blockwise rounds share a data window across four block jobs but repeat teacher inference and block-local backward passes; Appendix F accounts for this repeated work.

In the sparse Llama hybrid, blockwise training updates SSM mixers and readouts, plus available value/output projections in replaced layers. Retained attention, MLPs and norms stay fixed (Table 20); the pure-SSM regional control below has a broader trainable scope.

E.2 Does boundary placement determine the endpoint?

The three tested partitions retain the same attention layers. Each listed block end is inclusive; after the fixed layer-0 singleton, the next block starts immediately after the preceding end. Each partition has one training trajectory, with point estimates and no paired intervals. Their endpoint scores are close, with mixed directions across tasks (Table 14). Attention-aligned boundaries give the highest MMLU, while shifted and uniform boundaries give the highest Lambada and WinoGrande scores, respectively.

Table 14: Single-run endpoints after six blockwise-training rounds under three partitions of the same attention layout $\{0, 9, 17, 21\}$.

Partition	MMLU	Lamb.	Wino.	Hella.	PPL
Attention-aligned $0/9/17/21/27$	51.70	17.50	59.70	63.50	30.66
Shifted $0/11/19/23/27$	51.27	17.58	59.43	63.49	30.84
Uniform blocks $0/6/13/20/27$	51.06	16.86	59.83	63.45	30.74

E.3 Where does regional joint training recover performance?

Prior analysis of Llama identifies a gather-and-aggregate computation around layers 16–17 that is poorly preserved by independently distilled SSM replacements (Bick et al., 2025b). This motivates a specific prediction: joint adaptation near that region may be more useful than applying the same training to early or late regions. We test it in a pure-SSM Llama student, training one of three separated seven-layer regions, $[2, 8]$, $[11, 17]$, or $[22, 28]$, under a shared data window and training budget. Indices are zero-based and inclusive. All parameters within the active region train, including SSMs, readouts, MLPs, and norms; Table 20 summarizes the settings. Each evaluated model contains only one trained region; the remaining layers retain their initial weights. All start from the same layerwise-distilled Llamba-8B-untied-unaligned model, with no attention retained. Its direct evaluation before blockwise training gives 34.1 MMLU. Each region has one training trajectory; the experiment compares regional continuations with this shared starting model. Equal-budget continuation of independent fitting was not evaluated.

Joint training recovers capability beyond this starting point, with the largest MMLU gains in the predicted region (Figure 3). At the third common round, its gain is 3.8 percentage points, compared with 0.1 and 0.7 for the early and late regions. The location predicted from prior analysis thus yields the largest observed MMLU recovery. Recovery is task-dependent: the early region gives higher WinoGrande and lower perplexity at that endpoint (Table 15). Prior gather-and-aggregate analysis concerns in-context computations such as MMLU option association; the long-context value-retrieval tests in this paper probe a different behavior.

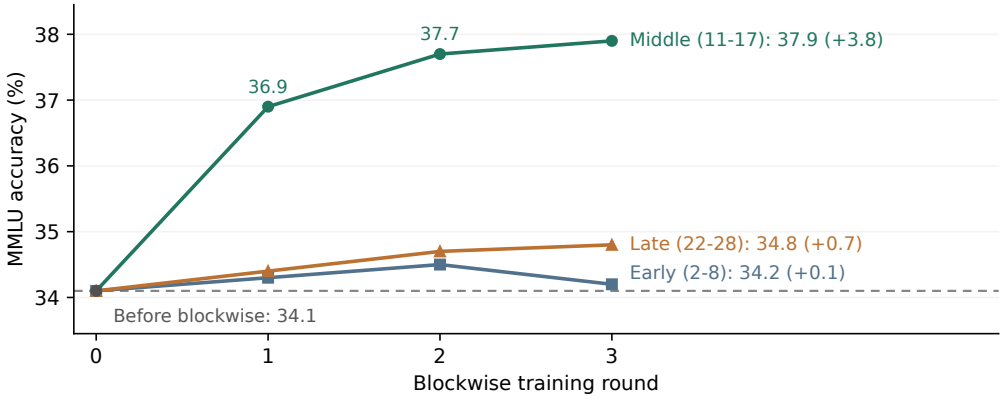


Figure 3: MMLU recovery from the common layerwise-distilled pure-SSM starting model. The dashed line marks its score before blockwise training; endpoint labels give gains in percentage points. The middle region contains the previously identified critical layers. Each region contains seven layers. Points trace successive rounds of each region's run. The vertical axis starts above zero.

Table 15: Single-run pure-SSM Llama region-training endpoints, including the common starting model and the third common round for each region. Figure 3 gives the full MMLU trajectories. Task scores are accuracy percentages; HellaSwag uses normalized accuracy.

Round	Trained layers	MMLU	Lamb.	Wino.	Hella.	PPL
0	None	34.1	20.2	63.4	74.0	29.37
3	2–8	34.2	20.6	66.2	74.0	28.43
3	11–17	37.9	21.2	63.8	74.6	28.96
3	22–28	34.8	20.9	63.9	74.1	28.97

F Global Training Paths and Boundary Supervision

The Qwen training-path study compares cumulative rounds 7–9 under the common teacher/data regime. Additional continuations from a common blockwise checkpoint test whether a boundary penalty remains useful during global refinement.

Prediction and boundary objectives. The global objective in Equation 5 uses

$$\mathcal{L}_{\mathrm{KL}} = \mathbb{E}_x \, \mathrm{KL}(p_{\mathrm{teacher}}(\cdot \mid x) \,\|\, p_{\mathrm{hybrid}}(\cdot \mid x)) \,, \tag{12}$$

where KL is Kullback–Leibler divergence and both distributions predict the next token from the same prefix. Optional boundary supervision gives

$$\mathcal{L}_{\mathrm{global+boundary}} = \mathcal{L}_{\mathrm{global}} + \lambda \mathcal{L}_{\mathrm{boundary,norm}}. \tag{13}$$

The normalized squared penalty below acts on the full student's complete layer outputs; it differs from local and blockwise unsquared L2.

F.1 Conditions of the common-regime path comparison

The main comparison pairs cumulative rounds 7, 8, and 9: direct global rounds 7–9 versus six blockwise rounds plus global rounds 1–3. Both start from the same locally prepared Qwen bank and retain attention at $\{0, 9, 17, 21\}$. All these rounds use the Qwen2.5-1.5B teacher and FineWeb. One round presents $32 \times 5000 \times 2048 = 327.68$M tokens. The shared data window is counted once for token exposure; each block job performs its own teacher inference and block-local computation.

Both global-training paths use KL plus 0.1 token cross-entropy, temperature 1, learning rate 2×10^{-5} with minimum 2×10^{-6}, a per-round cosine schedule with 0.05 warm-up and no constant-rate plateau, Adam coefficients $(0.9, 0.999)$, weight decay 0.1, and gradient clipping 1. Global refinement resumes after the six blockwise data windows, matching direct training's cumulative data offsets. Blockwise settings and parameter scopes are specified in Appendix E; these stage-specific differences are part of the tested training path.

Table 3 uses round-final full-MMLU and Lambada evaluations at cumulative rounds 7–9, without paired significance intervals. The common round-9 endpoint also supplies Table 4. MMLU uses the scoring rule in Appendix A.2; the saved-score tie analysis in Appendix I.5 covers the shared-bank study.

F.2 Boundary penalty in the Qwen training-path study

For the Qwen training-path study continuations, the base trainer hooks complete decoder outputs at zero-based boundaries $\{9, 17, 21\}$. Let $H_b^S(X)$ and $H_b^T(X)$ denote student and teacher layer-b outputs on batch X. Before optimization, it evaluates an initial batch X_0 and sets

$$c_b = \begin{cases} \|H_b^S(X_0) - H_b^T(X_0)\|_F^2, & \|H_b^S(X_0) - H_b^T(X_0)\|_F^2 > 10^{-8}, \\ 1, & \text{otherwise.} \end{cases} \tag{14}$$

Each training microbatch contributes $\sum_b \|H_b^S - H_b^T\|_F^2 / \max(c_b, 10^{-8})$. Differences are computed in float32, summed over batch, token, and hidden coordinates, with no extra token mask or mean

over boundaries. The scales remain fixed within each trainer invocation and are recalibrated at the start of each round in this study, using an optimization batch.

The initial penalty coefficient is calibrated to a boundary/KL gradient-norm ratio of 0.2. Fixed keeps this coefficient constant. Annealed halves its starting coefficient between successive rounds and decreases it within each round with a cosine schedule. All three shared-bank layouts reuse their initial scales and coefficients (Appendix I.2).

F.3 What does continuing boundary supervision preserve?

All three arms start from the same model after six blockwise-training rounds. The No penalty arm uses KL and token cross-entropy alone; Annealed and Fixed add scheduled or constant boundary supervision. Each arm has one training trajectory. Table 16 reports round-final point estimates at every common round. Fixed boundary supervision raises MMLU by 0.3–0.8 points, while its Lambada deficit narrows from 1.1 to 0.1 points.

Table 16: Six successive single-run endpoints per boundary objective in the Qwen training-path study. Values are accuracy percentages; MMLU uses micro accuracy.

Task	Objective	Round 1	2	3	4	5	6
MMLU	No penalty	50.5	50.3	50.7	50.2	50.2	50.2
	Annealed	50.6	50.3	50.5	50.5	50.6	50.0
	Fixed	51.0	50.9	51.0	50.7	51.0	50.9
Lambada	No penalty	46.8	48.7	49.8	50.2	50.2	50.1
	Annealed	46.1	48.4	49.8	49.9	50.5	50.7
	Fixed	45.7	48.0	49.1	49.5	50.0	50.0

Output and boundary objectives need not improve together. The training measurements in Table 17 compare Annealed with Fixed. In the first round, KL falls while the normalized boundary loss rises in both arms. Prediction and boundary losses therefore move in opposite directions during this part of training. The boundary values use round-specific normalization, so cross-round magnitudes reflect both model changes and recalibration.

Table 17: Training objectives for the two boundary schedules. Start is the first logged measurement in round 1; the other columns are round endpoints. Boundary is the normalized squared-error sum before multiplication by its penalty coefficient. Its normalization is recalibrated each round.

Loss	Objective	Start	Round 1	2	3
KL	Annealed	0.2189	0.1323	0.1292	0.1043
	Fixed	0.2189	0.1332	0.1294	0.1051
Boundary	Annealed	4.03	5.58	4.45	2.29
	Fixed	4.02	5.35	4.40	2.34

Uncertainty for the two boundary schedules. The reported Annealed-minus-Fixed bootstrap at round 6 gives $-0.77[-1.22, -0.33]$ for subject-macro MMLU and $+0.68[0.23, 1.13]$ for Lambada. These unadjusted intervals quantify the Annealed–Fixed comparison. Across rounds 4–6, eight of 12 task-round intervals contain zero. The MMLU intervals weight subjects equally; Table 16 reports micro accuracy.

G Qwen Recovery and Published Capability Context

The Qwen training-path study uses the $\{0, 9, 17, 21\}$ assembly, six blockwise rounds and three global rounds under the common teacher/data regime. This appendix adds perplexity, tasks not used for selection and published-model context to the main recovery results.

G.1 Additional capability and perplexity results

ARC-Easy/Challenge (Clark et al., 2018), PIQA (Bisk et al., 2020), and OpenBookQA (Mihaylov et al., 2018) were not used for layer selection. Blockwise training increases ARC-E, ARC-C and PIQA accuracy and decreases OpenBookQA accuracy (Table 18); the scores remain close to the teacher's point estimates. Perplexity also decreases after blockwise training while remaining above the teacher's.

Table 18: Additional-task accuracy before and after blockwise training. ARC-E/C, PIQA and OBQA all use ordinary accuracy (`acc`), with $N = 2376/1172/1838/500$, respectively. The final column adds $\mathrm{PPL}_{\mathrm{tok}}$ under the definition in Appendix A. Global-stage scores on these additional metrics are unreported.

Checkpoint	ARC-E	ARC-C	PIQA	OBQA	$\mathrm{PPL}_{\mathrm{tok}}$
Local-repair assembly	75.51	40.27	74.16	30.60	31.28
Blockwise, round 6	75.72	41.30	74.86	29.40	30.66
Teacher	75.25	40.78	75.95	31.00	27.08

G.2 Capability context across model families

Table 19 places the recovered models alongside published capability levels under each source's protocol. The published Llamba model and our experimental bank are separate checkpoints. Section 2 compares MOHAWK and Jet-Nemotron by design.

Table 19: Published capability context, with source-specific evaluation protocols. Values are percentages. Published model names indicate nominal scale; our Qwen hybrid registers 1.70B parameters (Appendix A). Rows differ in initialization, data, training and evaluation; they do not constitute a matched-budget comparison. The Qwen ReCompose row uses the common-regime endpoint in Table 4; the dash refers to normalized HellaSwag at that round-final checkpoint.

Model	Attention configuration	MMLU	Hella.	Wino.
Teacher references and converted models				
Qwen2.5-1.5B teacher	28/28 full	59.1	67.6	62.3
ReCompose Qwen-derived	4/28 full	50.7	–	62.9
Zebra-Llama-1B	4/16 MLA	37.91	58.93	58.88
Zebra-Llama-1B	8/16 MLA	38.94	60.54	61.25
Llama-3.1-8B-Instruct teacher	32/32 full	68.3	79.6	73.5
ReCompose Llama-8B	4/32 full	60.5	77.0	70.9
Zebra-Llama-8B	8/32 MLA	54.60	78.40	72.45
Llamba-8B	None	60.29	76.25	72.77
Mamba-in-Llama-8B	4/32 full	50.78	73.14	63.38
Mamba-in-Llama-8B	8/32 full	53.71	77.70	64.80
Architectures pretrained from scratch				
Hymba-1.5B	Hybrid heads	51.19*	53.55§	66.61
RecurrentGemma-2B	Local, window 2048	40.2	60.3	55.7
RecurrentGemma-9B	Local, window 2048	55.1	80.1	73.7
Zamba2-1.2B	Shared attention	43.1*	70.9†	68.0
Zamba2-7B	Shared attention	64.7	81.5	76.9
Samba-1.7B	Sliding-window attention	48.01	49.74‡	72.93

A separately saved Qwen global-round-3 checkpoint scores 64.6 on normalized HellaSwag; it is distinct from the round-final checkpoint in the table. MLA is multi-head latent attention, not retained full attention. *5-shot MMLU; †10-shot HellaSwag; ‡ordinary HellaSwag accuracy rather than normalized accuracy; §HellaSwag metric as reported in Hymba's source table. Other scores are zero-shot. Zebra and Llamba scores use the evaluation reported by Yang et al. (2025, Table 1); Llamba's architecture is described by Bick et al. (2025a). Mamba-in-Llama uses the Mamba2 rows of Wang et al. (2024, Table 3). RecurrentGemma and Zamba2-7B use the instruction-tuned reevaluations in Bick et al. (2025a, arXiv v1, Table 1), not their original reports' base-model evaluations. Hymba, Zamba2-1.2B and Samba use their original reports (Dong et al., 2025; Glorioso et al., 2024; Ren et al., 2025).

H LLAMA RECOVERY AND LAYER-SELECTION CONTROLS

The Llama study uses a Llamba-8B layerwise-distilled bank with attention retained at zero-based $\{9, 11, 13, 17\}$. Its recipe and construction controls are specified below. For the assembly, blockwise round 8, global round 3 and teacher rows in Table 4, $\mathrm{PPL}_{\mathrm{tok}}$ is 35.25, 27.96, 25.55 and 24.13, respectively.

The reported 9.104B-token conversion budget accounts for the reused bank using Llamba's published 0.5B-token matrix-orientation and 5B-token hidden-state-alignment budgets (Bick et al., 2025a), then adds eight blockwise plus three global rounds (approximately 3.604B token presentations), counting each shared blockwise data window once. This accounting applies to Llama, separately from Qwen's post-assembly token budget. The binding tests in Appendix J examine recovery on a more demanding task.

H.1 TRAINING SETTINGS AND CHECKPOINT SELECTION

Table 20 distinguishes the sparse-hybrid recovery recipe from the pure-SSM region experiment. Both reuse the layerwise-distilled Llamba-8B-untied-unaligned bank and use Llama-3.1-8B-Instruct as teacher. All continuations use FineWeb sequences of length 2,048, an effective batch of 32, and 5,000 updates per round: 327.68M token presentations. Block jobs share the round's data window, so this token count does not multiply by the number of blocks or measure their repeated computation.

Table 20: Llama training recipes. Ranges include both endpoints. The teacher is fixed throughout; all student parameters not listed as updated stay fixed.

Setting	Hybrid blockwise	Hybrid global	Pure-SSM regional
Initialization	Layerwise bank; attention at $\{9, 11, 13, 17\}$	All five blocks from blockwise round 8	Layerwise bank; no attention
Trained span	$[0, 9]$, $[10, 11]$, $[12, 13]$, $[14, 17]$, $[18, 31]$; one per job	Whole model computation	One of $[2, 8]$, $[11, 17]$, $[22, 28]$
Updated weights	SSM mixers, including their readouts; available value/output projections in replaced layers	SSM mixers and retained-attention value/output projections	All active-layer parameters: SSMs, readouts, MLPs, norms
Fixed weights	Retained attention, MLPs, norms	Attention query/key projections, MLPs, norms, embedding and prediction head	Every layer outside the active region; embedding and prediction head
Input / target	Teacher block-entry / block-exit states	Same tokens, own states / teacher next-token distribution	Teacher region-entry / region-exit states
Loss	Mean tokenwise unsquared L2	KL, temperature 1; no CE	Mean tokenwise unsquared L2
Learning rate	3×10^{-5} to 5×10^{-6}	10^{-5} to 10^{-8}	5×10^{-6} to 5×10^{-7}
Reported checkpoints	Round 8, the parent of global training	Round 3, after eight blockwise rounds	Untrained baseline and all three round-final checkpoints

Blockwise and regional schedules use 3% warmup and a 30% stable phase; global refinement uses 10% warmup and an 80% stable phase before decay. Each new round advances the data window. Global rounds restart optimizer state; their trainable scope includes retained-attention value/output projections. As the table specifies, retained attention participates in hybrid blockwise computation while staying fixed. Pure-SSM regional recovery instead reflects adaptation of complete layers, including MLPs and norms.

The blockwise row in Table 4 is the round-8 parent of global training. An earlier round-6 checkpoint has the path's highest recorded MMLU: 61.4, with HellaSwag 76.3, WinoGrande 71.0, Lambada 37.0 and $\mathrm{PPL}_{\mathrm{tok}}$ 28.12. Global refinement starts from round 8, so the main table follows that continuation.

H.2 Does one retained layer support every capability?

The construction controls show that a layer can strongly support MMLU without restoring long-context retrieval. These construction comparisons evaluate the assembled model before any further training. They start from `Llamba-8B-untied-unaligned`, with attention modules taken from Llama-3.1-8B-Instruct. Their MMLU screens use fewer examples than Table 4 and are reported separately.

Adding one attention layer. The first screen evaluates all 16 subsets of $\{9, 11, 13, 17\}$ on zero-shot MMLU with ten examples per subject. Averaging each layer's addition over its eight matched backgrounds gives gains of 0.53, 2.32, 4.52 and 18.16 points for layers 9, 11, 13 and 17, respectively. Layer 17's gain is positive in all eight backgrounds, ranging from 15.09 to 21.40 points. These additions change the attention count by one.

Moving the fourth attention layer. Table 21 holds $\{9, 11, 13\}$ fixed and varies the fourth position under the same MMLU screen. It changes layer placement without changing the attention count.

Table 21: Llama construction with layers $\{9, 11, 13\}$ fixed and one further attention layer retained. MMLU is a zero-shot screen with ten examples in each of 57 subjects. Values are reported accuracy percentages.

Fourth attention layer	15	16	17	18	19
MMLU	32.98	39.30	57.72	48.07	34.56

All 20 distinct configurations in the factorial and neighborhood tests score zero on 128 NIAH examples at each of 16k and 32k, using seed 20260801, greedy cached decoding and a 16-token output limit. Layer 17 therefore has a large MMLU effect in these backgrounds without restoring measured long-context retrieval. The location is consistent with prior analysis of MMLU-associated gather-and-aggregate computation (Bick et al., 2025b). Together, the two tasks show capability-specific sensitivity to retention.

H.3 Do selected layer indices transfer between models?

Only layer indices transfer; all weights remain Llama-derived. Transferring the Qwen-selected indices $\{0, 9, 17, 21\}$ into Llama gives 53.33 MMLU, compared with 56.56 for the Llama-selected $\{9, 11, 13, 17\}$, a 3.23-point difference. Both are Llama assemblies without further training. This separate zero-shot screen uses 25 examples in each of 57 subjects, rather than the ten above. Both layouts score zero NIAH at all six lengths from 1k to 32k, with 128 examples per length and seed 20260802. The Llama-selected positions preserve more MMLU accuracy at the same attention count. Both construction controls report point estimates for the tested layouts.

H.4 Does choosing layers for retrieval change the starting model?

Selecting attention layers for retrieval produces a different capability profile from MMLU selection (Table 22). Both assemblies use the untied, unaligned Llamba-8B bank with four Llama-3.1-8B-Instruct attention layers, before blockwise or global training. Genetic search maximizes generation accuracy on 64 fixed 1k NIAH discovery examples (seed 20260731), breaking ties by teacher-forced retrieval margin. Neither MMLU nor the evaluation bank enters this fitness.

Table 22: Llama assemblies selected for MMLU or retrieval. The retrieval row is selected by genetic search. MMLU uses 25 examples per subject (57 subjects), zero-shot; NIAH uses 128 examples per length.

Layout	1k	2k	4k	8k	16k	32k	MMLU
MMLU $[9, 11, 13, 17]$	0	0	0	0	0	0	56.56
Retrieval $[2, 15, 23, 27]$	96.88	96.09	90.62	84.38	67.19	50.00	24.14

Both rows use FlashAttention-2 (Dao, 2024), cached greedy decoding and a 16-token output limit on the seed-20260802 NIAH bank. The final candidate was fixed before its own evaluation, but this bank had already been used to assess intermediate candidates. It therefore provides a selected-model comparison rather than an untouched confirmation test.

Short-context selection does not identify the best long-context candidate. An earlier four-attention candidate, $\{1, 2, 15, 27\}$, scores 89.84% at 1k, 84.38% at 16k and 76.56% at 32k, with 23.51% MMLU. The final 1k-discovery winner instead scores 96.88%, 67.19% and 50.00% at those lengths, with 24.14% MMLU (Table 22). Its improvement at the short length accompanies lower accuracy at longer contexts. This reversal motivates checking the intended context lengths during selection.

I Qwen Shared-Bank Layout Study

This study follows three Qwen2.5-1.5B assemblies through selection, repair and fresh-example evaluation. All retain four attention layers and share the replacement bank specified in Appendix D.4.

I.1 Different selections expose different starting deficits

Table 5 shows the starting capability trade-off. Table 23 summarizes the selection objectives and reranking procedures for these four-attention-layer models.

Table 23: Selection objectives and reranking for models retaining four attention layers. Recorded deduplicated candidate pools contain 1,533 (MMLU), 286 (NIAH), and 355 (Mixed) layouts. The MMLU total combines multiple searches, inherited candidates and preliminary evaluations; these are realized pool sizes, not prescribed search budgets.

Objective	Full rerank	Selection rule
MMLU	30	Six searches without warm starts use 25 examples per subject; the frozen top 30 are reranked on full MMLU.
NIAH	30	Discovery uses 50 fixed 1k examples; candidates scoring at least 0.98 are reranked by mean 16k/32k accuracy.
Mixed	31	Genetic search maximizes normalized MMLU plus mean 16k/32k NIAH; rerank the proxy top 30 and single-objective references.

Discovery scores can misrank candidates even within a selection objective. The MMLU discovery winner $\{9, 14, 17, 21\}$ falls to rank 16 on full MMLU (48.03), while $\{9, 17, 21, 22\}$ wins the rerank at 50.04.

For layout a, $M(a)$ is MMLU accuracy and $N_{16}(a), N_{32}(a)$ are retrieval accuracies, expressed as fractions. Mixed selection maximizes:

$$S(a) = \frac{M(a)}{M_{\text{ref}}} + \frac{[N_{16}(a) + N_{32}(a)]/2}{N_{\text{ref}}}. \tag{15}$$

Both terms have weight one, without chance subtraction or clipping. The references are the single-objective winners remeasured under the candidate protocol, frozen separately for proxy evaluation and full reranking; full references are $M_{\text{ref}} \approx 0.5004$ and $N_{\text{ref}} = 0.905$.

The mixed search space contains $\binom{28}{4} = 20{,}475$ four-layer sets. It uses a population of 24 (eight sets each from MMLU, NIAH and pattern-based initialization), 24 generations, mutation rate 0.45, tournament size three, one elite and seed one. Every fitness score is evaluated anew. Proxy MMLU uses 25 examples per subject, zero-shot; proxy NIAH uses 32 per length, indices $[100, 132)$ of the 500-example banks.

What the common evaluation establishes. Full evaluation uses 14,042 MMLU examples and 500 NIAH examples at each length. NIAH uses SDPA attention, the incremental-cache adapter, greedy decoding, and a 128-token output budget. The teacher scores 58.98 MMLU and 100/100 NIAH.

The mixed model's gain over the NIAH-selected model is 11.47 MMLU points (paired 95% CI $[10.61, 12.29]$), accompanied by a 9.0-point loss at 32k NIAH ($[-11.6, -6.4]$). Pairing uses NIAH example identities and subject-stratified MMLU with sample-count weights. Proxy samples are included in the full evaluation banks, which also determine selection. The intervals condition on that selection; fresh-example tests follow in Appendix I.4.

I.2 Training conditions

All three layouts follow the same repair protocol, with common rules for block partitioning, checkpoint selection, and boundary-penalty calibration. Table 24 lists the resulting configurations. Blockwise training supplies teacher entry states, propagates student states within each block, and updates all active-block parameters against teacher exit states using token-averaged unsquared Euclidean distance.

Table 24: Configurations produced by the shared repair rules. Layer indices are zero-based; intervals are inclusive. Attention-only layer 22 in the MMLU layout stays fixed during blockwise training; all layers remain present. The coefficient λ is calibrated at the start of global training.

Selection	Retained layers	Trained blocks	λ
MMLU	$\{9, 17, 21, 22\}$	$[0, 9]$, $[10, 17]$, $[18, 21]$, $[23, 27]$	4.059×10^{-3}
NIAH	$\{6, 14, 19, 23\}$	$[0, 6]$, $[7, 14]$, $[15, 19]$, $[20, 23]$, $[24, 27]$	1.808×10^{-3}
Mixed	$\{6, 14, 19, 21\}$	$[0, 6]$, $[7, 14]$, $[15, 19]$, $[20, 21]$, $[22, 27]$	2.182×10^{-3}

All three train on FineWeb-Edu score-4 data (Penedo et al., 2024) for six rounds of 5,000 updates at sequence length 2,048 and effective batch 32. Each round uses a learning rate of 3×10^{-5}, with 3% warmup, a 30% constant-rate phase, and cosine decay to 5×10^{-6}. A fixed training-prefix probe selects the saved blocks within round six, under the common selection rule. In block order, the selected updates are 4,200/4,800/4,800/4,600 for MMLU, 4,400/4,600/5,000/4,800/4,800 for NIAH, and 5,000/5,000/5,000/4,600/5,000 for Mixed.

Each layout/objective configuration has one training trajectory. For each layout, global arms share this blockwise parent and run six further rounds with or without a normalized boundary penalty (*Fixed* and *No penalty*). All retain the 5,000-update rounds, length 2,048 and effective batch 32: 327.68M token presentations per round, or 1.966B per arm. The common base objective is $\mathcal{L}_{\mathrm{KL}} + 0.1\mathcal{L}_{\mathrm{CE}}$ at temperature one. Each global round starts a fresh optimizer, with 5% warmup and cosine decay from 2×10^{-5} to 2×10^{-6}. The first global round starts after the six blockwise data windows; both arms use the same progressive FineWeb-Edu offset rule across layouts and objectives. Global results in Figure 2 and Table 26 use round-six final checkpoints, not the blockwise probe-selection rule.

The penalty matches complete decoder-layer outputs at the retained-attention positions. Its coefficient, calibrated once to an initial boundary/KL gradient ratio of 0.2, is given in Table 24. All three save and reuse their first-round boundary normalization scales and coefficients across global rounds. The normalized squared-error form is defined in Appendix F, but that training-path study recalibrates its scales each round. Data offsets use approximate document skips. Within each layout, the global arms share a blockwise parent and differ by the calibrated boundary penalty. Applying the same calibration rule across layouts enables comparison of their observed responses to this procedure.

I.3 Recovery and the effects of continued boundary supervision

Recovery occurs at different stages for different tasks. The NIAH-selected model has a complete assembly-to-global evaluation (Table 25). HellaSwag improves by 13.83 points during blockwise training and a further 0.93 during global refinement. Lambada instead gains 2.62 and 28.92 points. SQuAD completion falls from 38.30 to 29.46 before reaching 45.58: the 16.12-point gain from the blockwise parent is a 7.27-point net improvement over assembly. Thus an improving final score can conceal an intermediate regression.

Table 25: Task-specific stages of recovery for the NIAH-selected layout. All values are percentages under the same evaluation protocol. Blockwise uses the selected round-six blocks; Global is its unpenalized round-six continuation. Norm. denotes normalized accuracy; contains denotes answer containment. The SWDE protocol is the version-0 Harness task used in Table 26. Percentage-point differences are computed before rounding.

Task	Assembly	Blockwise	Global
MMLU	32.53	37.04	39.24
HellaSwag (norm.)	49.15	62.98	63.91
PIQA (norm.)	68.23	75.35	75.08
WinoGrande	56.75	59.59	61.01
OpenBookQA (norm.)	36.20	40.20	40.00
Lambada accuracy	20.73	23.35	52.26
SQuAD completion (contains)	38.30	29.46	45.58
SWDE extraction (contains)	51.49	56.71	72.19
NIAH-32k	81.0	84.4	94.6

Table 26: Repair endpoints under the common shared-bank protocol. Each layout has one blockwise parent and two sibling global continuations: Global uses logit distillation, and Global+B adds the boundary penalty using the common calibration rule. Both end after six global rounds at update 5,000. Scores are percentages, except Lambada perplexity (PPL; lower is better). MMLU and the 500-example NIAH banks were also used during layout selection. Percentage-point differences are computed before rounding.

(a) Multiple-choice evaluation

Selection	Endpoint	MMLU	WinoGrande	HellaSwag norm.	OBQA norm.	PIQA norm.
MMLU	Blockwise	53.67	59.98	63.75	40.80	75.03
	Global	53.28	61.56	64.55	41.00	75.14
	Global+B	53.00	61.64	64.38	41.60	75.14
NIAH	Blockwise	37.04	59.59	62.98	40.20	75.35
	Global	39.24	61.01	63.91	40.00	75.08
	Global+B	38.93	61.33	64.07	40.00	74.81
Mixed	Blockwise	48.81	59.98	63.54	40.60	75.08
	Global	49.81	61.56	64.33	40.80	75.41
	Global+B	49.80	61.56	64.32	40.80	74.86

(b) Completion, extraction, and retrieval

Selection	Endpoint	Lambada acc.	Lambada PPL ↓	SQuAD contains	SWDE contains	NIAH 16k	NIAH 32k
MMLU	Blockwise	19.00	107.89	22.22	18.09	0.0	0.0
	Global	51.91	10.63	36.39	54.28	23.2	8.8
	Global+B	51.12	11.43	32.81	53.38	16.0	6.8
NIAH	Blockwise	23.35	55.08	29.46	56.71	99.8	84.4
	Global	52.26	9.80	45.58	72.19	100.0	94.6
	Global+B	51.72	10.03	44.57	74.08	100.0	96.6
Mixed	Blockwise	18.20	83.71	26.14	34.65	92.4	71.0
	Global	51.15	10.32	42.86	66.34	99.8	97.4
	Global+B	50.36	10.84	40.45	67.33	99.8	95.8

The evaluation contains 14,042 MMLU, 1,267 WinoGrande, 10,000 HellaSwag, 500 OpenBookQA (Mihaylov et al., 2018), and 1,838 PIQA (Bisk et al., 2020) examples. SQuAD completion (Rajpurkar et al., 2016) uses answer containment on 2,984 examples, rather than official EM/F1. SWDE (Hao et al., 2011) uses the version-0 LM Evaluation Harness task on 1,111 examples; its results are separate from the common-assembly extraction evaluation in Appendix I.4. Lambada

accuracy and PPL use 5,153 examples. Its PPL measures final-word continuation likelihood, distinct from the truncated-text, token-level $\mathrm{PPL}_{\mathrm{tok}}$ in Appendix A.

Similar completion scores can conceal different task profiles. Across the three layouts, unpenalized global training raises Lambada by 28.92–32.95 points, SQuAD completion by 14.18–16.72, and SWDE by 15.48–36.18 relative to blockwise training. WinoGrande and HellaSwag gains are smaller, while OpenBookQA and PIQA do not uniformly improve. The cross-layout Lambada accuracy range narrows from 5.14 to 1.11 points. Yet the global models still span 36.39–45.58 on SQuAD completion and 54.28–72.19 on SWDE. WinoGrande, HellaSwag, and PIQA endpoints are close, and were already close after blockwise training. The remaining variation in retrieval and extraction makes the intended task set important for model choice.

Training preserves some layout orderings and changes others. From blockwise to unpenalized global training, MMLU continues to rank the MMLU-selected model above Mixed and NIAH, while SQuAD completion and SWDE retain the reverse ordering. The NIAH/Mixed 32k retrieval ordering instead reverses: 81.0/72.0 at assembly and 84.4/71.0 after blockwise training become 94.6/97.4 after global refinement. Evaluating both construction and repair reveals which capability rankings persist under the common training protocol.

Boundary effects within each selected model. All contrasts below compare the penalized and unpenalized continuations from the same parent within each selection. On the MMLU-selected model, the penalty lowers NIAH at all 12 recorded round/length comparisons. After six rounds its differences are -7.2 points ($[-9.8, -4.6]$) at 16k and -2.0 ($[-3.6, -0.4]$) at 32k. For the NIAH-selected model, the final 32k difference is $+2.0$ ($[0.8, 3.4]$), and both arms score 100% at 16k. The mixed model instead loses 1.6 points ($[-2.8, -0.6]$) at 32k on the bank used for selection. The response to boundary supervision varies across layouts and tasks. At these endpoints, the penalty also lowers Lambada by 0.54–0.80 points and SQuAD completion by 1.01–3.59 across the three layouts. SWDE changes by -0.90, $+1.89$, and $+0.99$ points for MMLU, NIAH, and Mixed, respectively. Thus even within the NIAH-selected model, higher retrieval and extraction scores accompany lower Lambada and SQuAD completion scores. These additional task differences are point estimates.

MMLU differences are small relative to their uncertainty: penalty minus no-penalty is -0.285 points ($[-0.669, 0.093]$) for the MMLU selection, -0.306 ($[-0.669, 0.057]$) for the NIAH selection, and -0.014 ($[-0.313, 0.285]$) for mixed selection. All three intervals span zero and do not establish equivalence. The first two use the subject-stratified micro analysis in Appendix A.2.

Correct-answer transitions from the common parent. For the MMLU-selected model, the common blockwise parent answers no examples correctly on either recorded NIAH bank. Every correct answer after global training is therefore newly correct relative to this parent. Table 27 compares the two continuations using matched sample identities, prompts, and answers.

Table 27: Correct-answer overlap after six global-training rounds for the MMLU-selected model. Both continuations start from the same blockwise parent, which scores zero on these previously used evaluation banks. *Boundary only* denotes an answer correct only with the boundary term.

Length	N	Both correct	Logit only	Boundary only	Both wrong
16k	500	75	41	5	379
32k	500	31	13	3	453

The penalty yields fewer correct answers overall, but uniquely answers five 16k and three 32k examples. The sibling continuations thus recover partly different sets of answers from their common parent.

For the NIAH-selected model, the penalty's 32k advantage combines retention and recovery. Relative to the blockwise parent, training without the penalty newly answers 54 examples and loses three earlier correct answers; training with it newly answers 62 and loses one. Of the ten answers correct only with the penalty, two were already correct at the parent and eight were not.

I.4 RETRIEVAL ON NEW EXAMPLES

The initial retrieval ordering extends to new examples. To separate the observed profiles from reuse of the selection examples, we evaluate the three fixed assemblies on new 32k retrieval banks. For the original repeated-haystack format, a 500-example pool generated with Python seed 20260908 supplies the first 64 examples in each quarter of measured evidence depth. This selection uses no model scores and produces 256 examples with prompts and target answers distinct from those in the earlier comparison. Depth is tokens before the unique target occurrence divided by input tokens. An additional 256-example bank uses essay text in place of the repeated haystack. It changes one synthetic RULER variant.

Table 28: Retrieval before blockwise/global training on new 32k examples. Depth columns give correct counts out of 64 for each quarter of the original-format bank. Final columns give accuracy over 256 examples per bank. Earlier evidence is harder for both nonzero-scoring models.

	Evidence depth: earliest to latest				Accuracy (%)	
Selection	First	Second	Third	Last	Repeated filler	Essay text
MMLU	0	0	0	0	0.00	0.00
Mixed	6	50	60	63	69.92	42.58
NIAH	16	59	64	64	79.30	47.27

The ordering persists on both banks (Table 28), with earlier evidence markedly harder in the original format. Depth balancing changes the weighting from the older bank, so differences in absolute accuracy between banks do not isolate an effect of new examples. On our SWDE-derived extraction evaluation (Hao et al., 2011), MMLU/mixed/NIAH selection scores 14.67/34.65/53.11% over 1,111 rows. SWDE changes the task format and uses containment scoring, but its 295–1,938-token webpages test short-context extraction. These short webpages complement the synthetic long-context tests.

Mixed-model recovery persists, with an unresolved depth limit. On the new banks, the tested mixed-model global endpoints improve over their common blockwise parent, while all three fail in the earliest essay-text quarter (Table 6).

The boundary penalty ties the unpenalized endpoint on new original-format examples and leads by 2.73 points on the essay-text bank, opposite the direction on the original bank used for selection. The direction of the penalty's effect therefore varies across these task instances.

I.5 SENSITIVITY TO ANSWER SCORING

These checks reuse saved outputs to test how scoring rules affect the reported comparisons, without generating new model responses.

Retrieval scoring. The NIAH scorer credits a target appearing as a substring in the output. Among 72 disagreements between the two global objectives, wrong outputs often contain only a prefix of the target number: 54 of 62 cases for the MMLU selection and five of ten for the NIAH selection. The scorer can also accept extra digits: output `87022702` receives credit for target `8702270`. This motivates checking whether recovery persists under stricter number matching.

For the mixed model's fresh banks, we check this concern using all 1,536 saved predictions (two banks, three endpoints, 256 examples each). Requiring the seven-digit target to match a complete contiguous digit sequence, rather than part of a longer number, preserves every correctness label. Requiring it to be the first number changes one essay-text Fixed prediction from correct to incorrect: 171/256 becomes 170/256, while Blockwise and No penalty remain 138/256 and 164/256. The output starts with `14.` before giving the correct target `1499282`. These post-hoc checks preserve the recovery pattern under number-boundary and first-number scoring. They still allow explanatory text: none of the saved responses consists of the target alone.

MMLU likelihood ties. We examine saved option scores for the two shared-bank comparisons of global training with and without a boundary penalty. Their evaluations use Harness 0.4.11 and Transformers 4.49.0 with bfloat16 model outputs. Log-softmax and token summation retain that precision; conversion to Python float occurs after summation. Replaying the first-maximum rule reproduces all saved correctness labels. This output audit covers the shared-bank comparisons, not the training-path endpoints in Table 3. Despite small aggregate differences, the global arms disagree in correctness on 762 of 14,042 examples for the MMLU selection and 683 for the NIAH selection. The correct option ties with another option for the highest recorded score in at least one arm on 602 and 661 of these disagreements, respectively. These counts concern disagreement subsets, not all questions. Across all 14,042 questions, the no-penalty/penalty arms have 1,362/1,395 top ties for MMLU selection and 2,555/2,566 for NIAH selection. Only 159 and 22 disagreements have unique highest scores in both arms; one additional MMLU-selection disagreement has a nongold top tie. These exact ties exist before display rounding. We retain official accuracy and its first-maximum rule, but the ties limit interpreting answer changes as changes in knowledge or reasoning. The saved scores do not identify which computation produces the ties.

J Binding Recovery and Retained-Attention Tests

We test two aspects of binding recovery: discriminating an answer with all queried bindings correct, and changing answer preference when a relevant binding changes. Attention suppression then tests whether refinement changes the contribution of retained attention to these behaviors. The Llama recovery study uses Llama-3.1-8B-Instruct with retained attention layers $\{9, 11, 13, 17\}$ (zero-based). We compare its round-8 blockwise parent (BW) with checkpoints after three and five global rounds, denoted Global 3 and Global 5. BW is the initialization for global training and the blockwise checkpoint in Table 4.

J.1 Task, candidate difficulty, and evaluation

Our synthetic multi-query key–value (MQKV) task contains 32 bindings, four requested keys, values drawn from 1,000 possible value strings, and 16 filler lines per example. One version renders bindings in an explicit schema; the other uses templated dossier statements, such as a record specifying `value_0759` for `key_0021`. Each set contains 500 examples, independently generated with seeds 13 and 17. Thus comparisons between these two formats do not isolate wording on matched semantic worlds. The strict tests below use the dossier format.

We score a correct four-key JSON answer and eight alternative answers by summing teacher-forced continuation log probabilities. The model scores each continuation separately and does not receive a list of candidates in its prompt. The ordinary alternatives replace all four values with non-gold values from the context. Recovering any one binding can therefore separate the correct answer from every alternative. To remove this shortcut, the hard set preserves three correct values in each alternative and changes only one, using two alternatives for each queried field. All nine candidates have 48 scored tokens. This candidate-ranking test checks discrimination against an error at each queried field.

The ranking evaluations use the same 500 dossier examples, paired across models and interventions; the swap test uses a fixed subset described below. All reported 95% confidence intervals use 10,000 paired original-example percentile bootstrap resamples and are unadjusted. They condition on these inputs and one training trajectory; the corrected tests are identified separately in Appendix J.5.

J.2 Retained-attention intervention

For correct-answer score s_g and alternative scores s_d, define the candidate margin $M = s_g - \log \sum_{d \neq g} \exp(s_d)$. This measures separation from the pooled alternatives; ranking accuracy instead checks whether the correct answer has the highest individual score. We jointly scale all four retained attention outputs at their residual write sites, allowing subsequent layers to receive the changed states. Writing B_i for decoder layer i,

$$h_i(\alpha) = B_i\big(h_{i-1}(\alpha); g_i(\alpha)\big), \qquad g_i(\alpha) = 1 + \alpha \mathbf{1}\{i \in \mathcal{A}\}. \tag{16}$$

No weights are updated, and replacement mixers are not directly rescaled. The exploratory grid was $\alpha \in \{0, -0.25, -0.5, -1\}$, with zero reproducing the original model and -1 removing the retained

attention writes. The strict tests fix $\alpha = -0.25$, multiplying these writes by 0.75. Let $M_c(\alpha)$ denote the margin averaged over examples at checkpoint c. Its suppression cost is

$$C_c(\alpha) = M_c(0) - M_c(\alpha). \tag{17}$$

Positive C_c means suppression lowers mean margin; negative C_c means it raises mean margin. Neither sign alone determines the change in ranking accuracy.

J.3 Recovery under stricter candidate ranking

Table 29: Candidate ranking on the same 500 dossier examples. Suppression multiplies all four retained attention writes by 0.75 ($\alpha = -0.25$). $M_c(0)$ is the baseline mean hard-candidate margin; $C_c(-0.25)$ is its suppression cost (Eq. 17). Dashes mark the hybrid-specific intervention, which is not applied to the teacher.

	Ranking accuracy (%)			Hard-candidate margins	
Checkpoint	Ordinary	Hard	Hard, suppressed	$M_c(0)$	$C_c(-0.25)$
Teacher	100.0	90.2	–	$+4.878$	–
BW	25.2	4.0	3.4	-4.925	-0.553
Global 3	50.4	8.8	6.6	-2.988	-0.392
Global 5	58.2	11.8	8.2	-2.703	-0.240

Behavioral recovery. The teacher ranks the correct answer highest on 451/500 hard items, whereas BW, Global 3 and Global 5 do so on 20, 44 and 59. Uniform random choice among nine candidates has expected accuracy $1/9 \approx 11.1\%$. Relative to BW, Global 5 gains 7.8 percentage points in accuracy (95% CI $[4.6, 11.0]$) and 2.222 margin units ($[1.927, 2.528]$). Refinement therefore recovers some strict binding behavior, but the teacher gap remains much larger than the gain. Strong ordinary-candidate performance alone would substantially understate this remaining difficulty.

Suppression and the two metrics. The hard-set margin costs are negative at all three student checkpoints: BW -0.553 ($[-0.669, -0.438]$), Global 3 -0.392 ($[-0.489, -0.295]$), and Global 5 -0.240 ($[-0.341, -0.137]$). Suppression therefore raises the mean margin, yet lowers hard-set accuracy (Table 29). These metrics summarize different properties: separation from the pooled alternatives and the identity of the top-ranked answer.

J.4 Responses to changed evidence

We next hold two candidate answers fixed and change the context. Answer A is correct for the original context. A target swap makes B correct by exchanging the queried value with a non-target record's value. The unrelated control exchanges two non-target bindings, leaving A correct. Both edits exchange values between two records. A structurally valid subset of 64 dossier examples was fixed before evaluation, with 16 examples per changed query field, and used for all four models.

Let $D_c(x; \alpha) = s_c(A \mid x; \alpha) - s_c(B \mid x; \alpha)$. A correct A-to-B response requires $D_c(x; 0) > 0$ and $D_c(x_{\text{target}}; 0) < 0$. Table 30 reports these decisions and the direct target-minus-unrelated contrast.

Table 30: Evidence swaps on 64 structurally selected dossier examples, without attention suppression. Correct A-to-B responses require correct preferences in both the original and target-swapped contexts. $\Delta D = D(x_{\text{target}}; 0) - D(x_{\text{unrelated}}; 0)$; negative values favor B relative to the unrelated control. The last two columns report the mean contrast and its unadjusted 95% interval.

Checkpoint	Correct A-to-B responses	Mean ΔD	95% CI
Teacher	62/64	-20.467	$[-22.948, -17.917]$
BW	14/64	-1.027	$[-1.908, -0.220]$
Global 3	25/64	-1.370	$[-2.325, -0.496]$
Global 5	30/64	-1.809	$[-2.949, -0.776]$

Refined checkpoints more often follow the changed binding, although Global 5 still does so on fewer than half the examples. The teacher's 62 correct responses show that most swaps are interpretable under the common scorer. Its other two flips are in the wrong direction. The teacher's unrelated-minus-original margin change is $+0.012$ ($[-0.194, 0.231]$).

J.5 Attention contribution across training stages

The behavioral gains above are distinct from changes in the effect of attention suppression. For hard-candidate ranking, we compare the margin cost $C_c(-0.25)$ across checkpoints. For evidence swaps, define target sensitivity $T_c(\alpha) = D_c(x; \alpha) - D_c(x_{\text{target}}; \alpha)$, averaged over the paired examples. Its suppression cost $S_c = T_c(0) - T_c(-0.25)$ is 0.324, 0.620 and 0.824 at BW, Global 3 and Global 5. Table 31 separates the two planned Global 5-minus-BW comparisons from the behavioral recovery results.

Table 31: Changes in attention-suppression costs from BW to Global 5. The two planned contrasts form one Holm-corrected family. Confidence intervals are unadjusted paired-bootstrap intervals; adjusted p-values use a normal reference distribution, not the bootstrap.

Measure	Global 5 − BW	95% CI	Holm p
Hard-ranking margin cost C	$+0.314$	$[0.161, 0.463]$	1.29×10^{-4}
Swap target-sensitivity cost S	$+0.501$	$[-0.149, 1.147]$	0.133

The positive change in C means the negative cost becomes smaller; it is not a positive attention cost at Global 5. The swap comparison does not resolve a change in S. Subtracting the corresponding unrelated-swap interaction gives $+0.442$ ($[-0.203, 1.069]$), which also spans zero. The improved response to changed evidence therefore does not establish greater dependence on retained attention.

K Runtime Benefits of Sparse Conversion

To test the efficiency motivation for sparse conversion, we compare a four-attention-layer Llama hybrid with its Transformer teacher on one A100-80GB GPU (Table 32), using a construction-stage assembly. Prefill processes the input context before generation; decode timings below measure one attention or SSM operator during subsequent generation.

The hybrid has lower end-to-end prefill latency at every tested length, with speedups of 1.06–$1.54\times$. At 32k, latency falls from 7,832 to 5,087 ms. Individual SSM operators are faster than attention in both the prefill and decode measurements. Whole-model latency also includes retained attention and the other layer operations. These measurements support the efficiency motivation for sparse conversion and hence the value of recovering capabilities within the selected layout. They profile construction-stage weights; final repaired checkpoints and end-to-end decoding throughput are outside this measurement.

Table 32: Construction-stage single-A100-80GB profiling. Times are milliseconds. Total prefill is measured end to end; mixer prefill and decode columns give individual operator latency, not end-to-end throughput. Speedup is the teacher/hybrid prefill latency ratio; dashes mark unreported timings.

	Total prefill			Mixer prefill		Mixer decode	
Context	Teacher	Hybrid	Speedup	Attn.	SSM	Attn.	SSM
1k	408	325	1.26×	4.31	1.62	3.76	0.96
2k	526	471	1.12×	5.06	3.16	3.73	0.97
4k	829	779	1.06×	7.85	6.09	3.78	0.96
8k	1502	1403	1.07×	15.57	12.18	3.82	0.97
16k	3180	2696	1.18×	38.37	23.71	–	–
32k	7832	5087	1.54×	124.53	47.59	–	–