One Checkpoint, Many Sizes: Quality and Latency in Multi-Exit LMs

Anonymous authors
Paper under double-blind review

Abstract

Multi-exit language models turn one training run into a configurable family of deployment sizes. We characterize their quality and latency through capacity-controlled comparisons of chained Matryoshka suites with independently trained, token-matched model ladders. At the 200M scale, resizing our initially under-allocated later exits (16–31% below their baselines) closes the observed accuracy gap, with panel-dependent gains. At 3B, we evaluate a branched design that combines earlier junction taps with exit-private neck layers. A version within 0.5% of classic largest-exit capacity gains 0.37–0.84 accuracy points over classic in two paired seeds, with lower byte-level perplexity. In these runs, its seven-task accuracy approaches the independent vanilla endpoints and its byte-PPL improves over both. Timing makes the execution tradeoff explicit: deep exits take 22–47% longer per generated token at batch 1 on one H100 despite similar matmul FLOPs. Together, the results show how capacity allocation and branching can improve suite quality, while measured latency identifies the workload-specific cost of each exit. This integrated view of capacity, quality, and runtime provides a concrete basis for designing and selecting multi-exit model families.

1 Introduction

Language models serve applications with multiple resource budgets. An application may use a small model for interactive responses and a larger one when more computation is available. Multi-exit pretraining addresses this setting with a shared training artifact: one checkpoint exposes a ladder of usable models, each ending at a different exit. The central design question is how to realize this shared-training benefit while delivering useful quality and runtime at each deployment size.

Chained Matryoshka suites (Godey and Artzi, 2026) make this question particularly concrete. Their segments increase in width, and a junction passes each segment's representation to the next. Every exit is a complete language model; in the classic chain, each shallow execution path is contained in the deeper paths. This structure shares training across deployment sizes and couples two design choices. Dividing a depth budget across widths determines how much capacity each exit can use, while reaching a large exit may require more sequential layers than a standalone model of similar size. Evaluating capacity and execution together makes the value of each suite member explicit.

We ask: *after accounting for usable capacity, what quality and latency tradeoffs emerge, and how can relaxing strict nesting improve them?* We investigate this question at two pretraining scales using independently trained, token-matched model ladders. Our initial width-first 200M implementation provides an allocation contrast: its later exits are substantially smaller than their baselines. This mismatch belongs to our instantiation; Godey and Artzi (2026) already target per-exit capacity. Resized controls let us assess the quality recovered by correcting allocation and then study the residual gap at the larger scale.

For the residual gap, we study a *necked* alternative. Each exit can own private tail layers, while an earlier junction tap lets deeper exits bypass those layers. This relaxes strict nesting: the shared body remains reusable, but a shallow model's complete path need not be a prefix

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.

of a deeper model. Private exit depth itself has a direct precedent in Balcony (Jamialahmadi et al., 2025). Our intervention combines it with early taps during end-to-end pretraining and tests the resulting bundle under explicit capacity controls.

The evidence supports three connected findings:

- **Capacity-aware allocation recovers quality.** Resizing our under-capacity 200M exits removes their observed later-exit deficit. Alternative task panels and a depth-prefix control identify the exit- and task-specific scope of this recovery (section 4).
- **Relaxing nesting improves the 3B suites.** The necked design improves over classic in two paired runs. The largest-exit gain persists at closely matched capacity, bringing the suite closer to independent-model accuracy (section 5).
- **Measured runtime guides deployment choices.** Timing quantifies the batch-1 decoding cost at similar parameter counts and FLOPs. Higher-batch results show how geometry and utilization shape the tradeoff under each workload (section 6).

The contribution is a controlled empirical account of these tradeoffs, together with a tested branching design. Building on established capacity-matched evaluation, we assess the combined effect of private depth, early taps, and path geometry. Capacity controls, paired training runs, and actual timing connect the quality recovery to the concrete architectures and workloads in which it is observed.

2 Related Work

One network, multiple deployment sizes. Slimmable Networks, Once-for-All, and BigNAS learn shared supernets (Yu et al., 2019; Cai et al., 2020; Yu et al., 2020); comparisons to independently trained members are an established part of this line. Matryoshka representation learning similarly evaluates nested representations (Kusupati et al., 2022). In language models, depth-adaptive training, LayerDrop, and DynaBERT vary depth or width (Elbayad et al., 2020; Fan et al., 2020; Hou et al., 2020). SortedNet provides a general multi-dimensional framework, and Sorted LLaMA is the closer precedent for shared-head depth prefixes (Valipour et al., 2023; Kavehzadeh et al., 2024). MatFormer nests feed-forward subnetworks, while LLaMAFlex targets elastic width and depth (Devvrit et al., 2024; Cai et al., 2025). Our main study follows the chained, width-increasing suites of Godey and Artzi (2026). They target per-exit capacity and study loss reweighting for a residual largest-exit gap. Their capacity-targeted construction provides the reference point for our allocation audit and subsequent branching comparisons.

What exits share. Early-exit models span jointly trained exits and exits added to an existing model (Teerapittayanon et al., 2016; Xin et al., 2020; Pan et al., 2024; Chen et al., 2024). Balcony is the direct architectural precedent for exit-specific transformer layers: it trains those layers post hoc in a frozen trunk (Jamialahmadi et al., 2025). Our necked suites instead jointly pretrain the shared body and private layers, and change where the next segment taps the representation. The matched sibling tests the joint effect of these design changes. This differs from post-training model extraction or routing (Cai et al., 2024; Xia et al., 2024) and per-token compute allocation (Raposo et al., 2024). The depth-prefix control offers a structurally different comparison using an explicitly specified shared-head training recipe.

Fixed exits versus adaptive inference. CALM selects depth during generation (Schuster et al., 2022); LayerSkip, Draft & Verify, and EESD use exits or skipped layers in self-speculative decoding (Elhoushi et al., 2024; Zhang et al., 2024; Liu et al., 2024). Kangaroo trains a shallow draft adapter for a frozen model (Liu et al., 2024). These methods address execution policies, whereas our main measurements time a fixed selected exit. Adaptive serving systems such as HELIOS further optimize model and exit selection (Kumar et al., 2026). Likewise, query routing and answer deferral address different decisions: routing selects a model before execution, while a cascade may escalate after observing an answer (Chen et al., 2024; Ong et al., 2025; Ding et al., 2024; Dekoninck et al., 2025). Our supplementary offline deferral analysis characterizes prediction-level choices between exits. Conditional execution and

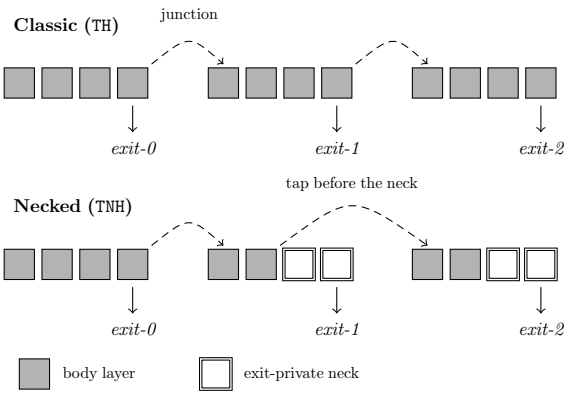


Figure 1: Classic and necked execution paths. A classic junction continues from the full preceding segment. An early tap instead bypasses an exit-private neck: the shallow exit uses the neck, while deeper exits continue from the shared body. This schematic follows the 200M ownership pattern; the 3B redesign also gives the first exit a private neck. Boxes illustrate ownership, not literal layer counts.

the cross-model KV transfer studied by Zhu and Yang (2025) are distinct systems-level comparisons.

3 Method

3.1 Classic and necked execution paths

A classic suite comprises segments $0, \ldots, K$ of non-decreasing hidden width. An exit after segment k consists of the path through that segment, an exit normalization, and a width-matched language-model head. A junction normalizes and rescales the preceding representation and concatenates a fresh embedding lookup on the additional width channels (Godey and Artzi, 2026). All exits train on the same token batch. Smaller exits use the largest exit's distribution as a distillation target:

$$L_m = 0.7\,\mathrm{CE}_m + 0.3\,\mathrm{Distill}_m \quad (m < K), \qquad L_K = \mathrm{CE}_K.$$

In the classic chain, shallow losses shape the shared prefix used by deeper exits; each loss directly updates the layers on its own path.

The necked design changes this ownership pattern (fig. 1). Within a segment, a shared body is followed by an exit-private neck. The local exit uses both, while the junction into the next segment taps the body before the neck. Deeper exits bypass that private tail in both their forward paths and loss computation. The design jointly changes private capacity, junction location, and active path depth. Our comparisons evaluate this complete architectural intervention.

We use *classic* and *necked* for the layouts denoted `TH` and `TNH` in the detailed run tables. The *matched necked* 3B sibling reduces the largest exit's private depth to bring its capacity within 0.5% of classic. It retains the necked layout at the smaller exits, whose capacities are reported separately rather than assumed to match classic. Full geometries and run-name mappings appear in appendices A and B.

3.2 Comparisons and capacity conventions

Each suite is compared with a *vanilla ladder*: independently trained models using the same tokenizer, data, optimizer, schedule, and per-model token budget. The nominal ladders are 50M/100M/200M at 20B training tokens and 500M/1.5B/3B at 35B tokens. These are scale labels, not exact parameter counts. The primary experiments use SmolLM2 tokenization and data (Allal et al., 2025), grouped-query attention, and untied heads. At 3B, the classic

geometry follows the published segment dimensions; at 200M, the initial width choices are our instantiation. Exact capacities are audited at model initialization (table 6).

We distinguish three quantities that answer different questions. *Exit-resident parameters* describe a separately deployed exit with untied heads. *Logical per-token parameters*, denoted NE+LH, count the non-embedding path plus the head used by that exit; this matters when exit heads are slices of a shared embedding table. *All-exits-hot storage* describes keeping the whole suite available. Each quantity answers a specific resource question: tied storage savings coexist with the exit's per-token head multiplication. The secondary, large-vocabulary tied-head experiments therefore use NE+LH for matching, with storage reported separately (appendices A and J.1).

The 200M comparisons proceed from the initial width-first chain to a capacity-resized classic chain and then a necked design at similar capacity. At 3B, we compare classic, necked, and matched necked using two paired seeds, and train two independent vanilla largest endpoints. The smaller vanilla models and most supporting arms have one seed. A separate shared-head depth-prefix control has three suite seeds at its largest exit. Its objective sums all exit cross-entropies on each batch, without distillation. Supporting junction, distillation, and neck-dose controls are retained in the appendix; the secondary tied family evaluates a combined change in tokenizer, data, and head treatment, extending the study to a second training configuration.

Capacity matching controls one source of variation while keeping geometry and optimization visible. Resizing changes depth, and a suite trains multiple exit objectives within its token budget. We therefore report the allocation ledger, analytical training-FLOP accounting, and deployment measurements separately (tables 5 and 6 and appendix J.1), providing a distinct reference for each resource comparison.

What the controls identify. The comparisons answer complementary questions. The initial-to-resized 200M comparison measures the outcome of changing depth and parameter allocation together. The classic-to-necked comparison evaluates an alternative layout through the layers it shares, the junctions it taps, and the paths each exit executes. The matched sibling then tests whether the largest-exit gain persists when its parameter budget is brought close to classic. This sequence locates the observed recovery within progressively more controlled architectural comparisons.

The independent ladder supplies the quality reference for models trained separately at each size. A positive necked-minus-classic difference measures recovery within the suite family, while suite-minus-vanilla differences locate that recovery relative to independent training. Small observed differences establish proximity for these checkpoints; equivalence would require a separate statistical analysis. Reporting both contrasts makes the architectural improvement and the standalone-model comparison directly interpretable.

3.3 Evaluation

Our main quality measure is mean accuracy on seven zero-shot tasks: ARC-Easy, ARC-Challenge, HellaSwag, LAMBADA, OpenBookQA, PIQA, and WinoGrande (Sakaguchi et al., 2020). This follows the task panel of Godey and Artzi (2026); the metric mapping is our explicit reproduction convention; the original paper leaves this mapping unspecified. We use normalized accuracy for ARC-Challenge, HellaSwag, OpenBookQA, and PIQA, and raw accuracy for the others. The appendix calls this the *paper metric*. LAMBADA contributes a word-prediction measure alongside the multiple-choice tasks.

We also report `avg8`, which replaces OpenBookQA with BoolQ and SciQ, and the arithmetic mean of byte-level perplexity on WikiText-103, PG19, arXiv, and PubMed Central. The alternative accuracy panel tests sensitivity to benchmark composition and retains near-chance cells. All task and corpus values, including additional 3B tasks, appear in appendices C and D. Accuracy differences are percentage points; lower perplexity is better. Individual training-seed values expose the observed run variation. Item- and document-bootstrap intervals in the appendix quantify evaluation-sample uncertainty conditional on fixed checkpoints, complementing those seed comparisons.

Table 1: Largest-exit 200M comparison after 20B tokens (seed 1234). Capacity is exit-resident parameters divided by vanilla's count. Accuracy panels are in percent. Resizing removes the initial deficit, but the necked advantage depends on the panel. Complete ladders and supporting controls are in appendix C.

Model	Capacity ratio	Seven-task accuracy	`avg8`
Initial width-first	0.694	41.25	—
Resized classic	1.000	43.48	51.40
Necked	0.999	44.04	51.10
Vanilla	1.000	42.82	51.23

The accuracy panels and byte-PPL offer complementary views of quality. An unweighted task mean gives each benchmark equal influence; replacing tasks can alter a small aggregate difference while leaving the model predictions unchanged. Byte-PPL measures predictive fit to held-out text, normalized by byte length to support comparisons across tokenizers. Its equal-corpus average gives each domain the same weight, distinct from pooling likelihood over all documents. Reporting both agreement and variation across these views exposes the task and domain composition of each model's performance.

4 Capacity allocation changes the 200M comparison

Capacity-aware resizing recovers the later-exit accuracy of our 200M chain. The initial width-first configuration allocates fewer parameters to these exits than the independent ladder and scores below it. Table 1 makes this mismatch visible at the largest exit. Deepening the classic chain to match capacity reverses the seven-task comparison; the necked design also exceeds vanilla on that panel at similar capacity. Because resizing changes both depth and parameter count, the result establishes recovery by this allocation change as a whole.

The two panels locate the quality recovery more precisely. On `avg8`, the necked largest exit is slightly below vanilla, and its smallest exit is sensitive to BoolQ. Several seven-task gains occur on near-floor tasks. Language modeling supplies a consistent complementary result: the necked largest exit improves byte-PPL on all four corpora, whereas the initial under-capacity chain trails vanilla (table 10). The combined evidence supports capacity-sensitive accuracy recovery and improved held-out language modeling for the resized necked checkpoint.

The supplementary controls characterize the response to other design changes. Context-gated junctions, removal of distillation, and a larger private neck show mixed or non-monotone changes relative to their matched parents (tables 9 and 11). These single-seed diagnostics provide a direct comparison of the tested alternatives and place the main redesign's gains in the context of the broader architecture study.

A structurally different control extends the allocation analysis. For a shared-head depth-prefix model, the natural ladder gives its smaller exits excess capacity while leaving its largest exit under-sized. Resizing brings the later exits close to their vanilla capacities. The resized largest exit's mean difference is near zero across three suite seeds, alongside a pronounced small-exit LAMBADA gap (table 7). Evaluating the full ladder captures both behaviors and identifies the task-specific quality available at each deployment size.

The depth-prefix control further distinguishes parameter count from path geometry. Its resized smallest member has more parameters than the standalone reference but uses a much shallower path, placing its generative gap outside a simple parameter-shortfall explanation. Geometry and joint training remain coupled, while the largest-exit mean characterizes the other end of the same ladder. Together with the chained-suite audit, this control motivates a per-exit assessment that follows capacity matching with direct task evaluation. That assessment makes the smallest, intermediate, and largest members visible in the architecture-selection decision.

Table 2: Largest-exit 3B quality after 35B tokens. Each pair lists seeds 1234 / 3456; accuracy is in percent and byte-PPL is the four-corpus mean. Capacity is relative to vanilla. Matched necked is within 0.5% of classic capacity. All exits and paired differences appear in table 12. Both training runs are shown individually, making the observed variation explicit.

Model	Capacity ratio	Seven-task accuracy	`avg8`	Byte-PPL
Classic	1.006	55.17 / 55.59	62.22 / 62.14	1.7813 / 1.7840
Necked	1.037	56.29 / 56.03	63.45 / 62.95	1.7721 / 1.7705
Matched necked	1.011	56.01 / 55.96	63.15 / 63.22	1.7713 / 1.7721
Vanilla	1.000	56.11 / 56.17	62.91 / 63.69	1.7770 / 1.7745

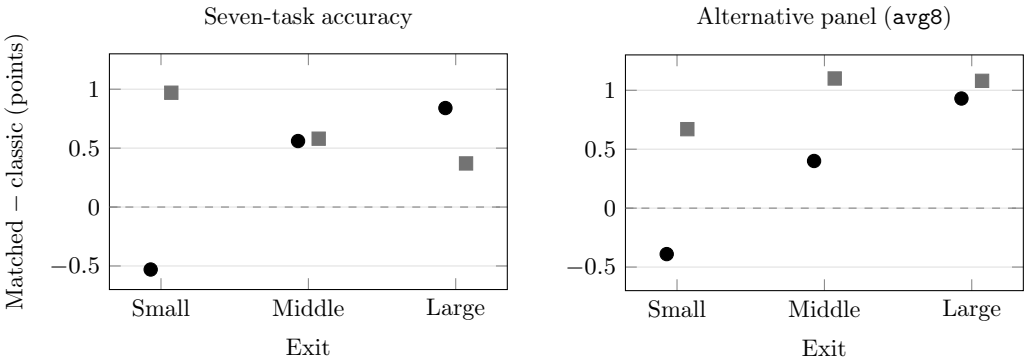


Figure 2: The matched redesign across the full 3B ladder. Circles/squares denote seeds 1234/3456. Each marker is a paired training-seed difference, not a confidence bound. Middle and largest exits improve on both panels in both seeds; the smallest exit changes sign. Capacity matching refers to the largest exit; the middle exit has fewer parameters than classic, and the smallest has the same count. Exact scores and capacities are in tables 6 and 12.

5 A branched redesign recovers residual quality at 3B

At 3B, the matched-capacity comparisons identify a clear opportunity for architectural improvement. Both classic largest-exit checkpoints score below both independent vanilla endpoints on the seven-task panel and `avg8` (table 2); their byte-PPL is also slightly higher. Classic already has comparable parameter capacity. This setting therefore lets us evaluate how the branching design improves quality beyond the initial allocation correction.

The redesign improves quality in both paired runs. With otherwise identical training recipes, the necked arm scores above classic at every exit on both accuracy panels and has lower byte-PPL in both paired seeds. Tasks and exits are correlated measurements of these two training pairs. The original necked largest exit also has more capacity than classic; the matched sibling provides the direct follow-up comparison at a closely aligned parameter budget.

The matched sibling addresses this ambiguity. It reduces the largest exit's private depth while retaining the branching design. At nearly the same largest-exit capacity as classic, it gains 0.84 and 0.37 points on the seven-task panel in the two paired seeds. Both `avg8` and byte-PPL favor the matched sibling as well. The middle-exit comparison also favors the redesign, while the smallest-exit difference changes sign (table 12). The resulting claim is specific: the bundled redesign recovers quality over the observed classic checkpoints, including under a close largest-exit capacity match.

The full ladder identifies exit-specific gains. Figure 2 places the largest-exit recovery alongside the smaller members. The matched redesign's middle exit improves with fewer parameters than classic, and its direction agrees across the two panels. The smallest exit has the same parameter count and changes its relative ordering between seeds. These paired results

Table 3: Batch-1 decoding relative to the corresponding vanilla model on one H100 (TP=1, bf16, context length 2048). The latency range spans pooled medians for eager, fused, and FlashAttention-2 backends, each measured over three randomized process restarts. This descriptive backend range complements the matched-cost comparison; it is not a confidence interval.

Model	Exit	Matmul-FLOP ratio	Per-token latency ratio
Classic	Middle	0.964	1.22–1.24
	Largest	0.975	1.41–1.43
Necked	Middle	0.938	1.23–1.26
	Largest	1.001	1.45–1.47
Matched necked	Middle	0.938	1.22–1.24
	Largest	0.973	1.45–1.46

distinguish the consistent middle- and largest-exit gains from the smallest exit's variability, supporting selection at the deployment size an application will actually use.

The independent ladder locates the quality tradeoff. The matched largest exit is slightly below both vanilla checkpoints on the seven-task panel, falls between them on `avg8`, and improves over both on byte-PPL. Two runs support a checkpoint-level comparison rather than an equivalence or superiority claim. Showing both vanilla runs also separates panel sensitivity from training-run variation. The result is a concrete combination of nearby seven-task accuracy and improved held-out language modeling.

The task breakdown identifies the level at which the recovery persists. WinoGrande dominates the classic deficit in the first seed but not the second, and the early prediction of a positive necked-minus-classic WinoGrande difference changes sign. The aggregate improvement remains present across both seeds while its task composition changes (table 17 and appendix E). The paired results thus support the redesign's aggregate recovery across these two distinct task-level patterns.

Supporting diagnostics extend the architectural picture. The large-vocabulary tied-head arm agrees directionally: its necked bundle improves over classic, with a residual gap to vanilla. Untying heads and late schedule changes retain that gap; adding deep-path capacity narrows it while changing several factors at once (tables 13, 14, 18 and 23). These one-seed comparisons assess combined interventions in head treatment, optimization, and geometry. Together, they place the paired branching result within a broader set of tested designs.

6 Measured latency reveals the execution tradeoff

Measured runtime complements the quality comparisons by pricing each selected exit. We time fixed exits against their corresponding standalone models on one H100 at TP=1 in bf16. The primary batch-1 comparison uses three randomized process restarts and three attention backends. We verify cached decode against prefill logits, count matmul FLOPs consistently on both sides, and report absolute timings and restart-level summaries in tables 27 and 28.

The restarts characterize execution variability within the named hardware and software setting. Within each fresh process, backend-block, model-boot, and exit order are randomized; CUDA-event measurements follow warm-up iterations. These are repeated timings, distinct from training-seed or hardware replication. Pooling them gives a descriptive summary, while restart-level medians and within-process quantiles in table 28 expose sensitivity to process initialization. This protocol supports the reported direction and scale of the suite-to-vanilla latency difference across the evaluated attention backends.

Timing quantifies the cost of the recovered quality. Across all three primary suite designs, middle and largest exits take longer than their vanilla counterparts (table 3), including when their matmul FLOPs are lower. The matched necked model retains essentially the same largest-exit timing as the original necked model, locating this cost beyond that arm's extra

Table 4: Cost of obtaining the whole 3B ladder at the same per-model token budget (seed 1234). The suite is the original necked arm, not the matched sibling. FLOPs use the stated matmul-only approximation; checkpoint directories contain resumable model and optimizer state. Inference residency and end-to-end training time are separate resource quantities. Full accounting is in table 32.

Resource	One necked suite	Independent ladder
Training runs	1	3
Resumable checkpoint storage (GB)	242.9	274.8
Training matmul FLOPs (EF)	≤ 950	976

capacity. Fused batch-1 prefill gives the same qualitative ordering. The measurements attach an explicit runtime cost to the quality comparisons.

The execution paths provide an architectural context for this measurement. Classic and matched necked largest exits traverse 39 and 35 layers, respectively, compared with 28 in the standalone model. Similar matrix-computation totals are distributed across more sequential operations and different matrix shapes. Individual junction and kernel contributions remain coupled in these timings. The observed pattern is consistent with depth and utilization shaping execution, making measured runtime a useful complement to parameter and FLOP accounting.

Workload-specific measurements guide deployment. The batch sweep in appendix I shows that some exits approach or beat vanilla prefill latency at high batch sizes as utilization rises, while others retain a runtime premium. These measurements and the tied-family timings use a single-start protocol, so they provide exploratory workload comparisons alongside the stronger randomized batch-1 evidence. Largest-batch allocation failures remain marked as unmeasured cells. The measured cells identify the workloads in which FLOP advantages translate into improved prefill throughput.

Fixed-exit measurements inform adaptive serving. A suite offers multiple quality tiers from a shared training artifact, and analytical accounting can favor it over training a full ladder (appendix J.1). Conditional serving additionally involves an execution policy, residency choices, and reuse of shared computation. Our offline deferral study maps accuracy–escalation tradeoffs from saved outputs; conditional latency remains outside its measurements (appendix J). Agreement-based selection uses both predictions, whereas margin-based selection uses the small exit's output. These distinctions specify the decisions and costs an adaptive implementation would evaluate.

7 What one checkpoint buys

One checkpoint offers a resource benefit when the objective is to obtain and maintain a family of models. A single-size deployment compares one exit directly with one standalone model; a deployment that uses the whole ladder also accounts for obtaining all its members. The appropriate training reference follows that distinction: one independent run for a single required size, or the full set of independent runs for a multi-size deployment.

Table 4 summarizes the resource accounting for the original necked suite and its independent ladder. Checkpoint size measures resumable training state, including the optimizer, separately from inference weight memory. Training cost uses a matmul-only forward/backward approximation, with GPU-hours, energy, and operations outside that convention left unmeasured. Summing all deployed exit paths counts the shared body repeatedly, giving an upper bound within this accounting model. Stating these units explicitly makes the artifact and training comparisons reproducible.

Under these conventions, the suite offers a smaller resumable artifact and a favorable analytical training-FLOP bound for the whole ladder. Its original necked largest exit is larger than vanilla, and replicated quality differences remain panel-dependent, so the resource totals describe a

tradeoff across concrete model families. Together with the matched quality comparisons, they quantify the shared-artifact benefit alongside the performance of its members.

The comparison also makes the architectural choice explicit. Private necks provide exit-specific computation during training and fixed-exit inference. For continuation into a deeper exit, reuse is restricted to the shared body, with private-tail state discarded. Applications prioritizing a single training artifact can weigh that structure against the measured fixed-exit cost; interactive-decoding workloads can compare directly with standalone models. The resource and timing results support these distinct deployment decisions using the same trained suite.

8 Scope and limitations

Replication and intervention. The primary 3B comparisons contain two seeds per arm, including the vanilla largest endpoint. Smaller vanilla references, 200M chained arms, and tied-family diagnostics are mostly single-seed; the depth-prefix largest-exit control has three suite seeds. This supports observed paired gains at checkpoint level, with expected-effect and equivalence inference outside the available replication. The second matched gain and the two-seed mean fall inside the earlier descriptive neutral band (appendix D). Since resizing and branching change geometry together, we attribute recovery to the tested bundles. Individual runs and explicit contrasts keep that evidence fully visible.

Metrics and generality. Task-panel sensitivity is material, especially near chance at 200M. MMLU is near chance at these budgets and is reported separately from both summary panels. Primary PG19 results use a fixed 20-book sample; a 50-book paired audit supports the largest-exit ordering conditional on the tested checkpoints (table 21). Document-level intervals are available for that audit only. The additional depth-prefix recipe provides a second architectural setting, while the complete task and corpus tables expose the scope of each finding.

Deployment and data. Timing covers one H100, the recorded software stack, and fixed exits; cross-device KV transfer and end-to-end conditional latency remain unmeasured. Public web-derived corpora can contain personal, sensitive, biased, or copyrighted material, and these inherited risks remain relevant to downstream use. The study uses existing public datasets and benchmarks, with no new human-subject data or annotations. The reported configurations and workload-specific measurements define the setting in which the deployment findings can be assessed.

9 Conclusion

Capacity-aware allocation and branched execution offer complementary ways to improve multi-exit model families. Our 200M controls recover quality through resizing, and the 3B neck-and-tap redesign improves over classic in two paired runs at closely matched largest-exit capacity. Independent-model comparisons locate that recovery: seven-task accuracy approaches vanilla, while byte-PPL improves in the matched runs.

Measured runtime completes the picture. The batch-1 decoding premium and workload-dependent prefill results identify the execution cost of each quality tier. Alongside the smaller resumable artifact and analytical training-FLOP bound, these measurements make the shared-checkpoint tradeoff concrete. The study provides tested architectural choices and an integrated evaluation approach for building multi-size language-model families around explicit quality and deployment requirements.

10 Reproducibility statement

All runs use Megatron-LM with preserved configuration dumps per arm. Suite and vanilla runs share tokenizers, data, optimizer, schedule, and token budget; the suite additionally trains multiple exit objectives. Training seeds are 1234 throughout, plus 3456 for the second smollm2 3B classic/necked pair, the matched sibling, and the vanilla 3B endpoint, and 2345/3456 for the depth-prefix replications. The core per-exit parameter and FLOP ledger and its analytic reconstructions are in table 6. Offline evaluation uses frozen corpora and per-model tokenizers. Configuration dumps, raw per-task evaluation JSON, latency measurements, and scripts for the task panels, byte-PPL, accounting, and latency tables are preserved for release with the submission.

AI use statement

AI assistants were used for code refactoring and evaluation-infrastructure debugging, for manuscript editing and proofreading, and to run an internal adversarial review simulation during drafting. All experimental designs, training runs, analyses, and scientific claims are the authors' own; no AI system generated experimental results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

L. B. Allal et al. SmolLM2: When smol goes big — data-centric training of a small language model. In *COLM*, 2025.

Apple Inc. Apple intelligence foundation language models. *arXiv preprint arXiv:2407.21075*, 2024.

H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han. Once-for-all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020.

R. Cai, S. Muralidharan, G. Heinrich, et al. Flextron: Many-in-one flexible large language model. In *ICML*, 2024.

R. Cai, S. Muralidharan, H. Yin, Z. Wang, J. Kautz, and P. Molchanov. LLaMAFlex: Many-in-one LLMs via generalized pruning and weight sharing. In *ICLR*, 2025.

L. G. G. Charpentier and D. Samuel. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning (CoNLL-BabyLM)*, pages 238–252, Singapore, 2023. Association for Computational Linguistics.

L. Chen, M. Zaharia, and J. Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

Y. Chen, X. Pan, Y. Li, B. Ding, and J. Zhou. EE-LLM: Large-scale training and inference of early-exit large language models with 3D parallelism. In *ICML*, 2024.

J. Dekoninck, M. Baader, and M. Vechev. A unified approach to routing and cascading for LLMs. In *ICML*, 2025.

Devvrit, S. Kudugunta, A. Kusupati, et al. MatFormer: Nested transformer for elastic inference. In *NeurIPS*, 2024.

D. Ding, A. Mallick, C. Wang, et al. Hybrid LLM: Cost-efficient and quality-aware query routing. In *ICLR*, 2024.

M. Elbayad, J. Gu, E. Grave, and M. Auli. Depth-adaptive transformer. In *ICLR*, 2020.

M. Elhoushi, A. Shrivastava, D. Liskovich, et al. LayerSkip: Enabling early-exit inference and self-speculative decoding. In *ACL*, 2024.

A. Fan, E. Grave, and A. Joulin. LayerDrop: Reducing transformer depth on demand with structured dropout. In *ICLR*, 2020.

N. Godey and Y. Artzi. Matryoshka language model suites. *arXiv preprint arXiv:2608.09703*, 2026.

L. Hou, Z. Huang, L. Shang, X. Jiang, X. Chen, and Q. Liu. DynaBERT: Dynamic BERT with adaptive width and depth. In *NeurIPS*, 2020.

B. Jamialahmadi, P. Kavehzadeh, M. Rezagholizadeh, et al. Balcony: A lightweight approach to dynamic inference of generative language models. In *EMNLP*, pages 24853–24867, 2025.

P. Kavehzadeh, M. Valipour, M. Tahaei, A. Ghodsi, B. Chen, and M. Rezagholizadeh. Sorted LLaMA: Unlocking the potential of intermediate layers of large language models for dynamic inference. In *Findings of EACL*, pages 2129–2145, 2024.

S. Kim, K. Mangalam, S. Moon, et al. Speculative decoding with big little decoder. In *NeurIPS*, 2023.

A. Kumar, S. Nag, J. Clemons, L. John, et al. HELIOS: Adaptive model and early-exit selection for efficient LLM inference serving. In *MLSys*, 2026.

A. Kusupati, G. Bhatt, A. Rege, et al. Matryoshka representation learning. In *NeurIPS*, 2022.

Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023.

J. Li, A. Fang, G. Smyrnis, et al. DataComp-LM: In search of the next generation of training sets for language models. In *NeurIPS*, 2024.

F. Liu, Y. Tang, Z. Liu, et al. Kangaroo: Lossless self-speculative decoding via double early exiting. In *NeurIPS*, 2024.

J. Liu, Q. Wang, J. Wang, X. Cai, et al. Speculative decoding via early-exiting for faster LLM inference with Thompson sampling control mechanism. In *Findings of ACL*, 2024.

Y. Lyu, B. Irukulapati, J. P. Champati, et al. Speculation at a distance: Where edge-cloud speculative decoding actually pays off. *arXiv preprint arXiv:2606.25091*, 2026.

I. Ong, A. Almahairi, V. Wu, et al. RouteLLM: Learning to route LLMs with preference data. In *ICLR*, 2025.

X. Pan, Y. Chen, Y. Li, et al. EE-Tuning: An economical yet scalable solution for tuning early-exit large language models. *arXiv preprint arXiv:2402.00518*, 2024.

D. Raposo et al. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

K. Sakaguchi, R. L. Le Bras, C. Bhagavatula, and Y. Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *AAAI*, pages 8732–8740, 2020.

T. Schuster, A. Fisch, J. Gupta, M. Dehghani, D. Bahri, V. Q. Tran, Y. Tay, and D. Metzler. Confident adaptive language modeling. In *NeurIPS*, 2022.

S. Teerapittayanon, B. McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *ICPR*, 2016.

M. Valipour, M. Rezagholizadeh, H. Rajabzadeh, et al. SortedNet: A scalable and generalized framework for training modular deep neural networks. *arXiv preprint arXiv:2309.00255*, 2023.

Y. Venkatesha, S. Kundu, and P. Panda. Fast and cost-effective speculative edge-cloud decoding with early exits. *arXiv preprint arXiv:2505.21594*, 2025.

M. Xia, T. Gao, Z. Zeng, and D. Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *ICLR*, 2024.

J. Xin et al. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *ACL*, 2020.

J. Yu, L. Yang, N. Xu, J. Yang, and T. Huang. Slimmable neural networks. In *ICLR*, 2019.

J. Yu, P. Jin, H. Liu, G. Bender, P.-J. Kindermans, M. Tan, et al. BigNAS: Scaling up neural architecture search with big single-stage models. In *ECCV*, 2020.

J. Zhang, J. Wang, H. Li, et al. Draft & verify: Lossless large language model acceleration via self-speculative decoding. In *ACL*, 2024.

P. Zhu and T. Yang. CE-LSLM: Efficient large-small language model inference and communication via cloud-edge collaboration. *arXiv preprint arXiv:2505.14085*, 2025.

A Arm nomenclature

Decodes every run-label token, evaluation panel, and capacity convention used in the main text.

Runs are named *family–layout–scale–tying*: family `Smol2` (SmolLM2 tokenizer, 49k vocabulary) or `G4` (a fixed 262,144-entry tokenizer run label, with a prefix-tied master table); layout `T` = shared trunk, `N` = exit-private neck layers (section 3.1), `H` = a width-matched LM head at every exit; a trailing *tied*/*untied* tag says whether those head slices are cut from one master table or independent; suffixes *-jcat* (ctx-gated junctions) and *-nodistill* (deep-exit distillation off) name the factorial variants. Thus `G4-TNH-3B-tied` = the necked 3B arm on the tied 262k-vocabulary master table. **Evaluation panels**: the *paper metric* is Godey and Artzi (2026)'s 7-task zero-shot panel; `avg8` adds boolq and sciq (raw `acc` for sciq) and drops openbookqa (task-mean), while `avg8-pool` is the same 8 tasks doc-weighted (answer-deferral analysis, appendix J); byte-PPL averages four OOD corpora under each model's own tokenizer. **Capacity conventions**: *exit-resident* (abbreviated "stored" in tables) counts the unique tensors that must be resident to run one selected exit, excluding other exits' private tensors; *suite storage* is instead the union of all checkpoint tensors and is not used for ladder matching; a server that keeps several exits hot must hold that union resident (for tied arms, the full master table), so we list it separately in table 6. *Active/in-use* counts the subset executed on that exit's forward path; it equals exit-resident mass for the untied arms reported here. For tied arms we judge *NE+LH*, the per-token parameter mass multiplied by a running exit: formally $\mathrm{NE{+}LH}(e) = P_{\text{nonemb trunk}}(e) + Vd_e$, where Vd_e is the exit's master-table head slice. Per-token prefill FLOPs are proportional to NE+LH up to attention-score terms and are independent of batch size as a count, not as elapsed time.

B Training configuration

All main comparison arms (suites, necked redesigns, vanilla ladders) share one optimizer recipe — seed 1234 (seed-3456 replicates for the smollm2 3B classic/necked arms and the matched sibling), bf16 mixed precision, no per-arm hyperparameter retuning; only the data-parallel micro-batch differs to fit memory, and deliberate perturbations (cooldown probes, loss-weighting variants) are named per arm in the main text.

Table 5: Capacity allocation by family. Percentages compare exits e0/e1/e2 with vanilla unless only e1/e2 are shown; counts are stored parameters except where NE+LH is specified. Short layout names map to the full run names in appendix A. No new runs are introduced.

Layout	Allocation	Capacity relative to vanilla
200M SmolLM2		
Initial (retired)	Implicit; depth first	Stored −16%/ − 31% (e1/e2)
Classic (TH)	Explicit; ladder-derived	Stored 48.8/123.2/256.7M; vanilla 48.8/124.4/256.7M
Factorial	Explicit; ladder-derived	`-jcat`/`-nodistill`: at TH-200M capacity
3B SmolLM2, untied		
Classic (TH)	Explicit; ladder-derived	Stored 0%/ − 1.7%/ + 0.6%
Classic + ctx-gate	Explicit; ladder-derived	Stored 0%/ − 1.7%/ + 0.6% (as TH)
Necked (TNH)	Explicit; trunk-lean	Stored 0%/ − 3.9%/ + 3.7%
Matched necked	Explicit; classic-proximate re-dose	Stored counts (suite / vanilla): e0: 478,200,832/478,200,832 e1: 1,368,878,336/1,424,081,088 e2: 3,035,958,016/3,004,316,160
3B G4, 262k vocabulary		
Classic, tied	Explicit; NE+LH	0%/ − 0.5%/ − 1.5%
Necked, tied	Explicit; trunk-lean	0%/ − 2.4%/ + 1.2%
Necked, payback	Explicit; deeper trunk (confounded)	In-use 0%/ − 4.2%/ + 3.9%
Classic, untied	Explicit; ladder-derived	NE+LH 0%/ − 0.5%/ − 1.5% (as tied classic). Exit-resident 0%/ + 6.6%/ + 10.2% from the 262k junction and head tables.

Table 6: Exact per-exit capacity and model-cost ledger. Each exit has separate parameter and FLOP rows; ratios are suite / vanilla. The 3B integers come from boot-time audits. The 200M entries are analytical reconstructions, cross-checked against available boot counts; retired-arm FLOPs are inferred from geometry. Accounting definitions and reconstruction details follow the table.

Exit (layers)	Quantity	Suite	Vanilla	Ratio
smollm2, V=49,152*, untied (exit-resident params); vanilla ladder 24L@1,024 / 28L@1,728 / 28L@2,560*				
SMOL2-TH-3B-UNTIED				
e0 (24)	Params	478,200,832	478,200,832	1.000
	FLOPs/token	1,056,964,608	1,056,964,608	1.000
e1 (34)	Params	1,400,339,712	1,424,081,088	0.983
	FLOPs/token	2,964,062,208	3,074,457,600	0.964
e2 (39)	Params	3,022,204,672	3,004,316,160	1.006
	FLOPs/token	6,184,632,320	6,343,884,800	0.975
SMOL2-TH-3B-JCAT				
e1 (34)	Params	1,400,798,464	1,424,081,088	0.984
	FLOPs/token	2,964,979,712	3,074,457,600	0.964
e2 (39)	Params	3,023,482,624	3,004,316,160	1.006
	FLOPs/token	6,187,188,224	6,343,884,800	0.975
SMOL2-TNH-3B-UNTIED				
e1 (32)	Params	1,368,878,336	1,424,081,088	0.961
	FLOPs/token	2,884,370,432	3,074,457,600	0.938
e2 (36)	Params	3,115,588,864	3,004,316,160	1.037
	FLOPs/token	6,352,535,552	6,343,884,800	1.001
SMOL2-TNH-3B-MATCHED				
e1 (32)	Params	1,368,878,336	1,424,081,088	0.961
	FLOPs/token	2,884,370,432	3,074,457,600	0.938
e2 (35)	Params	3,035,958,016	3,004,316,160	1.011
	FLOPs/token	6,174,408,704	6,343,884,800	0.973
262k, V=262,144 *(NE+LH; vanilla NE+LH 645,972,992 / 1,707,196,608 / 3,423,746,560, stored 914,408,448 / 2,160,181,440 / 4,094,835,200)*				
G4-TH-3B-TIED				
e0 (24)	Params	645,972,992	645,972,992	1.000
	FLOPs/token	1,493,172,224	1,493,172,224	1.000
e1 (33)	Params	1,698,196,224	1,707,196,608	0.995
	FLOPs/token	3,767,402,496	3,810,557,952	0.989
e2 (37)	Params	3,371,498,240	3,423,746,560	0.985
	FLOPs/token	7,256,539,136	7,434,403,840	0.976
G4-TNH-3B-TIED				
e1 (31)	Params	1,666,734,848	1,707,196,608	0.976
	FLOPs/token	3,687,710,720	3,810,557,952	0.968
e2 (34)	Params	3,464,882,432	3,423,746,560	1.012
	FLOPs/token	7,424,442,368	7,434,403,840	0.999

Continued on next page

Table 6 (continued): exact capacity and model cost

Exit (layers)	Quantity	Suite	Vanilla	Ratio
G4-TNH-3B-TIED-PAYBACK				
e1 (29)	Params	1,635,273,472	1,707,196,608	0.958
	FLOPs/token	3,608,018,944	3,810,557,952	0.947
e2 (31)	Params	3,558,266,624	3,423,746,560	1.039
	FLOPs/token	7,592,345,600	7,434,403,840	1.021
G4-TH-3B-UNTIED **(exit-resident)**				
e1 (33)	Params	2,302,176,000	2,160,181,440	1.066
	FLOPs/token	3,767,402,496	3,810,557,952	0.989
e2 (37)	Params	4,512,348,928	4,094,835,200	1.102
	FLOPs/token	7,256,539,136	7,434,403,840	0.976
200M smollm2, V=49,152, untied (tensor-count model for params; FLOPs from the same per-layer formula, the convention calibrated on the 3B boot audit); vanilla 5L@384 / 10L@640 / 14L@896 = 48,811,776 / 124,367,360 / 256,696,832				
all 200M arms				
e0 (5)	Params	48,811,776	48,811,776	1.000
	FLOPs/token	75,595,776	75,595,776	1.000
retired width-first (5/5/4; analytic FLOPs)				
e1 (10)	Params	104,704,000	124,367,360	0.842
	FLOPs/token	188,416,000	238,223,360	0.791
e2 (14)	Params	178,045,952	256,696,832	0.694
	FLOPs/token	339,279,872	528,023,552	0.643
SMOL2-TH-200M **(5/8/9)**				
e1 (13)	Params	123,139,840	124,367,360	0.990
	FLOPs/token	241,008,640	238,223,360	1.012
e2 (22)	Params	256,701,952	256,696,832	1.000
	FLOPs/token	548,995,072	528,023,552	1.040
SMOL2-TNH-200M **(5/6+2/10)**				
e1 (13)	Params	123,139,840	124,367,360	0.990
	FLOPs/token	241,008,640	238,223,360	1.012
e2 (21)	Params	256,455,424	256,696,832	0.999
	FLOPs/token	545,357,824	528,023,552	1.033

Ledger definitions and reconstruction notes. Params = exit-resident parameters (untied arms) or NE+LH (tied arms; trunk prefix plus the exit's master-table slice); FLOPs/token at S=2,048 = 2× matmul weights (norm vectors excluded) + attention-score matmuls $4\,S\,H\,d_h$ per layer ($QK^\top$ and PV, full context). Per-layer parameters are $2D^2+2D\,G\,d_h+3DF+2D$ (G KV groups, SwiGLU with three $D\times F$ matrices, two RMSNorm vectors); an untied exit adds the input table VD_0, each crossed junction delta $V(D_k-D_{k-1})$, its own head VD_j, and the exit's small final norm D_j (counted by the boot audit; the closed form reproduces the exit-0 integer exactly), while a tied exit's input, deltas, and head are slices of one master table so its resident mass equals NE+LH.

Suite storage (union of all tensors; boot-measured training-run parameter totals for all six 3B suites): smollm2 classic 3,185.8M, necked/matched 3,469.9M; 262k tied classic 3,371.5M, tied necked 3,655.6M, tied payback 3,939.7M, 262k untied 5,384.8M.

The 200M tensor-count model is validated to ≤0.9M against six measured 3B rows, and the measured 200M boot counts quoted in the text differ from it by <0.2M. The 200M `-jcat` variant adds two 128-rank context-gated junction bottlenecks — $2\times128\,(D_{\text{below}}+2\Delta D)$ parameters and three small GEMMs per token at each crossed junction (+262,144 params at 200M, +1,277,952 at 3B; both boot-verified against the training logs' parameter totals) — so `Smol2-TH-3B-jcat` matches the classic rows at e0 and adds the crossed junction's gate at e1/e2; `-nodistill` is tensor-identical to its parent (loss-coefficient change only); `neck4` moves two seg-1 traversed layers into the private tail at constant exit-1 parameters (per-exit geometry in appendix B).

Both scales share: AdamW $(0.9, 0.95)$, $\epsilon\,10^{-8}$, weight decay 0.01, grad clip 1.0, peak LR 4×10^{-4} (floor 4×10^{-5}), WSD schedule, bf16, seq. 2,048, global batch 512 sequences. 200M: 20B tokens, 19,073 steps, exits at hidden 384/640/896 (6/10/14 heads); suite arms `Smol2-TH-200M` (5L@384 + 8L@640 + 9L@896) and `Smol2-TNH-200M` (5 shared, then 6 shared + 2 private neck @640, then 10 exclusive @896; *neck4* moves two seg-1 traversed layers into the private tail at constant exit-1 parameters), plus the `-jcat`/`-nodistill` factorial at the same capacity; vanilla ladder 50M (5L@384), 100M (10L@640), 200M (14L@896). 3B: 35B tokens, 33,378 steps, staircase 24L@1,024 + 10L@2,304 + 5L@4,352, heads 16(64)/24(96)/34(128), RoPE, RMSNorm, SwiGLU; smollm2 arms `Smol2-TH-3B-untied` (the staircase above), `Smol2-TNH-3B-untied` (22sh+2nk @1,024, 8sh+2nk @2,304, 6 exclusive @4,352; exits 478M/1,369M/3,115M in use vs. the ladder's 478/1,424/3,004, i.e. 0%/−3.9%/+3.7%), a capacity-matched necked sibling (checkpoint suffix `suite_hydra_matched`: 22sh+2nk @1,024, 7sh+3nk @2,304, 6 exclusive @4,352; exits 478M/1,369M/3,036M in use, i.e. 0%/−3.9%/+1.1%; trained on the same recipe and reported as `Smol2-TNH-3B-matched` in the quality and latency tables), `Smol2-TH-3B-jcat` (classic geometry + ctx-gate); 262k-vocabulary tied redesigns `G4-TH-3B-tied` (seg 24/9/4, 37 layers stored) and `G4-TNH-3B-tied` (22sh+2nk @1,024, 7sh+2nk @2,304, 5 exclusive @4,352, 38 layers stored), both with the single [262,144 × 4,352] master table as shared embedding and LM head. The `G4-TH-3B-tied` trunk (seg 24/9/4) is the unique integer optimum of the NE+LH target, found by an integer sweep scored with a tensor-exact parameter model validated to <0.1M against the sibling suites' checkpoints. Vanilla ladders 500M (24L@1,024), 1.5B (28L@1,728), 3B (28L@2,560), one per tokenizer arm. Tokenizers/ data: SmolLM2 (49,152) on `smollm2_100b`; fixed 262k tokenizer on `dclm_blend` (Li et al., 2024) (g4 arm). Distillation coefficient $\alpha^d = 0.3$ (student exits only), matching the paper's best sweep value. Training-loss curves for the 200M arms, the 3B smollm2 arms, and the 262k-vocabulary redesigns appear in figs. 4 to 6.

C Per-benchmark 200M results

Table 7: Shared-trunk depth-prefix control. Panel A reports exact-count capacity differences relative to vanilla. Panel B gives seed-1234 paper-metric accuracy (%); parentheses are suite-minus-vanilla points. Panel C retains every largest-exit seed difference and the reported mean.

A. Capacity difference from vanilla

Arm (exit depths)	Exit-0	Exit-1	Exit-2
Natural (5/10/14)	+113.6%	+32.2%	−17.1%
Resized (1/7/18)	+14.9%	+3.2%	+1.6%

B. Accuracy at seed 1234

Arm	Exit-0	Exit-1	Exit-2
Vanilla ladder	35.55	39.93	42.82
Natural	38.94 (+3.39)	42.15 (+2.22)	43.36 (+0.54)
Resized	33.52 (−2.03)	39.97 (+0.04)	43.01 (+0.19)

C. Largest-exit accuracy difference by seed

Arm	1234	2345	3456	Mean
Natural	+0.54	−0.28	−0.32	−0.02
Resized	+0.19	+0.25	−0.44	−0.00

Table 8: 200M suite accuracy after 20B tokens (SmolLM2), split by evaluation panel. Cells are accuracy (%) and suite-minus-vanilla differences in points; bold marks the best suite value per column. Classic = TH; necked = TNH; ctx-gate = `-jcat`; no distill. = `-nodistill`. The neck4 arm doubles the middle exit's private tail at constant capacity. The retired arm has no archived `avg8` evaluation; paper Table-5 references provide seven-task accuracy only.

A. Seven-task accuracy

Model	Exit-0	Exit-1	Exit-2
Classic	35.97 (+0.42)	**40.61 (+0.68)**	43.48 (+0.66)
Necked (neck2)	35.97 (+0.42)	40.27 (+0.34)	**44.04 (+1.22)**
Necked (neck4)	35.34 (−0.21)	40.42 (+0.49)	43.31 (+0.49)
Classic + ctx-gate	**36.41 (+0.86)**	39.95 (+0.02)	42.76 (−0.06)
Necked + ctx-gate	35.75 (+0.20)	40.07 (+0.14)	43.32 (+0.50)
Classic, no distill.	35.00 (−0.55)	40.22 (+0.29)	43.31 (+0.48)
Necked, no distill.	35.42 (−0.12)	40.02 (+0.09)	43.59 (+0.77)
Vanilla ladder	35.55	39.93	42.82
Retired width-first	36.06 (+0.51)	39.27 (−0.66)	41.25 (−1.57)
Paper Table-5 suite	36.1	39.5	42.1
Paper Table-5 vanilla	36.2	39.1	42.0

B. Alternative panel (`avg8`)

Model	Exit-0	Exit-1	Exit-2
Classic	43.48 (−0.17)	48.41 (+0.81)	51.40 (+0.16)
Necked (neck2)	41.94 (−1.71)	47.52 (−0.08)	51.10 (−0.13)
Necked (neck4)	43.52 (−0.12)	**48.51 (+0.91)**	50.59 (−0.64)
Classic + ctx-gate	**43.92 (+0.27)**	47.67 (+0.07)	51.02 (−0.21)
Necked + ctx-gate	43.77 (+0.12)	48.03 (+0.43)	51.64 (+0.41)
Classic, no distill.	43.33 (−0.32)	47.91 (+0.31)	**51.74 (+0.51)**
Necked, no distill.	43.25 (−0.40)	47.96 (+0.36)	51.69 (+0.46)
Vanilla ladder	43.65	47.60	51.23
Retired width-first	—	—	—
Paper Table-5 suite	—	—	—
Paper Table-5 vanilla	—	—	—

Table 9: Suite-200M per-benchmark accuracy; each cell is `Smol2-TH-200M`/`Smol2-TNH-200M`/vanilla. Paper-metric convention: `acc_norm` on arc-c/hs/obqa/piqa, raw acc on arc-e/lambada/wg; boolq uses `acc`; sciq is shown both as `acc_norm` (the value in the 9-task avg row) and as raw `acc`, the value the `avg8` panel uses, so `avg8` in table 8 can be recomputed from the bold rows minus openbookqa plus boolq and raw sciq. Chance floors in parentheses: arc/hs/obqa/sciq 25%, piqa/wg/boolq 50%, lambada ≈0. Bold task rows form the paper-metric 7-task panel (7-task means and `avg8` in table 8); the **avg** row is the unweighted mean of the 9 rows above it.

	accuracy (%, cells TH/TNH/Van)		
Task (floor)	50M (exit-0)	100M (exit-1)	200M (exit-2)
arc-e (25)	43.69/43.43/44.99	50.84/50.59/49.62	53.96/55.68/54.97
arc-c (25)	22.53/22.53/22.01	24.91/24.57/25.00	27.47/26.88/26.37
hellaswag (25)	27.57/27.83/27.50	31.98/32.48/31.52	36.65/37.98/36.72
lambada (≈0)	20.24/20.40/19.85	29.52/29.92/28.12	35.05/36.21/34.33
openbookqa (25)	27.00/26.00/26.80	31.00/30.00/29.40	32.80/33.60/30.20
piqa (50)	60.23/59.85/59.30	64.25/63.98/63.76	66.92/66.81/67.30
winogrande (50)	50.51/51.78/48.38	51.78/50.36/52.09	51.54/51.14/49.88
boolq (50)	58.13/43.58/61.83	57.89/53.94/56.97	59.48/54.28/61.59
sciq (25)	56.0/58.7/57.0	68.4/67.1/65.4	70.8/71.4/69.2
avg (9 tasks)	**40.66/39.34/40.85**	**45.62/44.77/44.65**	**48.30/48.22/47.84**
sciq, raw `acc` (`avg8` input)	64.9/66.1/65.3	76.1/74.3/73.7	80.1/79.8/78.7

Table 10: 200M OOD byte-PPL per corpus (smollm2 arm, 20B tokens): suite/vanilla ($\Delta\%$ = (suite − vanilla)/vanilla; negative = suite better). Blocks: `Smol2-TH-200M`, `Smol2-TNH-200M`, and the retired classic-capacity suite. **avg** = arithmetic mean over the 4 corpora (the aggregation used in the main text). Evaluation sets: 1,000 wikitext-103 docs (111,491 scored tokens), 20 pg19 books (1,848,616), 40 arxiv docs (325,006), 200 pubmed-central docs (873,487), each scored with the model's own tokenizer.

Corpus	50M	100M	200M
Smol2-TH-200M			
wikitext-103	2.3655/2.3532 (+0.52%)	2.0638/2.0917 (−1.33%)	1.9513/1.9663 (−0.76%)
pg19	2.7656/2.7734 (−0.28%)	2.4689/2.4684 (+0.02%)	2.3255/2.3316 (−0.26%)
arxiv	2.3191/2.3037 (+0.67%)	2.0736/2.0937 (−0.96%)	1.9792/1.9936 (−0.72%)
pubmed_central	2.1527/2.1356 (+0.80%)	1.9053/1.9227 (−0.91%)	1.8157/1.8275 (−0.65%)
avg	**2.4007/2.3915** (+0.39%)	**2.1279/2.1441** (−0.76%)	**2.0179/2.0298** (−0.58%)
Smol2-TNH-200M			
wikitext-103	2.3441/2.3532 (−0.39%)	2.0624/2.0917 (−1.40%)	1.9369/1.9663 (−1.50%)
pg19	2.7644/2.7734 (−0.32%)	2.4569/2.4684 (−0.46%)	2.3119/2.3316 (−0.84%)
arxiv	2.3050/2.3037 (+0.06%)	2.0698/2.0937 (−1.14%)	1.9619/1.9936 (−1.59%)
pubmed_central	2.1424/2.1356 (+0.32%)	1.9025/1.9227 (−1.05%)	1.8045/1.8275 (−1.26%)
avg	**2.3890/2.3915** (−0.10%)	**2.1229/2.1441** (−0.99%)	**2.0038/2.0298** (−1.28%)
retired classic-capacity suite			
wikitext-103	2.3448/2.3532 (−0.36%)	2.1199/2.0917 (+1.35%)	2.0214/1.9663 (+2.80%)
pg19	2.7805/2.7734 (+0.26%)	2.5307/2.4684 (+2.53%)	2.4193/2.3316 (+3.76%)
arxiv	2.3151/2.3037 (+0.49%)	2.1170/2.0937 (+1.11%)	2.0333/1.9936 (+1.99%)
pubmed_central	2.1470/2.1356 (+0.53%)	1.9431/1.9227 (+1.06%)	1.8692/1.8275 (+2.28%)
avg	**2.3968/2.3915** (+0.22%)	**2.1777/2.1441** (+1.57%)	**2.0858/2.0298** (+2.76%)

Corpus-weighting sensitivity of the byte-PPL aggregate. The **avg** row weights four corpora of very different sizes equally (wikitext-103: 111,491 scored tokens vs. pg19: 1,848,616). Pooling byte-level cross-entropy over all scored tokens (token-weighted) instead moves the 200M-exit suite−vanilla aggregate for `TH`/`TNH`/the retired suite from $-0.58\%/-1.28\%/+2.76\%$ to $-0.43\%/-1.02\%/+3.17\%$, and leave-one-corpus-out reweight-

ing spans $[-0.66\%, -0.40\%]$, $[-1.36\%, -1.02\%]$, and $[+2.30\%, +3.17\%]$: magnitudes are weighting-sensitive at this scale, while every tested weighting preserves the sign and cross-arm ordering.

Table 11: 200M factorial and neck-dose arms: OOD byte-PPL, arithmetic mean over the four corpora, by exit; parentheses give the arm-to-parent contrast (arm − parent)/parent in %, i.e., the intervention effect rather than the arm-to-vanilla delta. Vanilla ladder: 2.3915 / 2.1441 / 2.0298.

Arm	50M (exit-0)	100M (exit-1)	200M (exit-2)
`Smol2-TH-200M` (parent)	2.4007	2.1279	2.0179
`Smol2-TH-200M-jcat`	2.3930 (−0.32%)	2.1276 (−0.01%)	2.0187 (+0.04%)
`Smol2-TH-200M-nodistill`	2.4208 (+0.84%)	2.1293 (+0.07%)	2.0223 (+0.22%)
`Smol2-TNH-200M` (parent, neck2)	2.3890	2.1229	2.0038
`Smol2-TNH-200M` (neck4)	2.4028 (+0.58%)	2.1258 (+0.13%)	2.0082 (+0.22%)
`Smol2-TNH-200M-jcat`	2.3951 (+0.26%)	2.1238 (+0.04%)	2.0059 (+0.10%)
`Smol2-TNH-200M-nodistill`	2.4143 (+1.06%)	2.1307 (+0.37%)	2.0037 (−0.00%)

D 3B PER-TASK ACCURACY AND OOD BYTE-PPL

Table 12: Core 3B comparison after 35B tokens, split by accuracy panel. Cells are accuracy (%); parentheses compare with the seed-1234 vanilla ladder, the pre-registered reference. Contrast rows give within-seed differences, not accuracy levels. The second vanilla run covers the largest endpoint only. Classic = `Smol2-TH-3B-untied`; necked = `Smol2-TNH-3B-untied`; matched = `Smol2-TNH-3B-matched`, within 0.5% of classic largest-exit capacity. All three suite arms use identical recipes at seeds 1234 and 3456. Values and differences retain the two-decimal per-task aggregation convention of `aggregate.py`.

A. Seven-task accuracy

Model / paired contrast	Seed	Exit-0	Exit-1	Exit-2
Classic	1234	47.44 (+0.46)	53.22 (−0.11)	55.17 (−0.94)
Classic	3456	46.36 (−0.62)	53.00 (−0.33)	55.59 (−0.52)
Necked	1234	47.57 (+0.59)	53.99 (+0.66)	56.29 (+0.18)
Necked	3456	46.59 (−0.39)	54.20 (+0.87)	56.03 (−0.08)
Necked − classic	1234	+0.13	+0.77	+1.12
Necked − classic	3456	+0.23	+1.20	+0.44
Matched necked	1234	46.91 (−0.07)	53.78 (+0.45)	56.01 (−0.10)
Matched necked	3456	47.33 (+0.35)	53.58 (+0.25)	55.96 (−0.15)
Matched − classic	1234	−0.53	+0.56	+0.84
Matched − classic	3456	+0.97	+0.58	+0.37
Vanilla	1234	46.98	53.33	56.11
Vanilla	3456	—	—	56.17

B. Alternative panel (`AVG8`)

Model / paired contrast	Seed	Exit-0	Exit-1	Exit-2
Classic	1234	55.31 (+0.88)	60.92 (+0.19)	62.22 (−0.69)
Classic	3456	54.39 (−0.04)	60.37 (−0.36)	62.14 (−0.77)
Necked	1234	55.41 (+0.98)	61.07 (+0.34)	63.45 (+0.54)
Necked	3456	54.73 (+0.30)	61.12 (+0.39)	62.95 (+0.04)
Necked − classic	1234	+0.10	+0.15	+1.23
Necked − classic	3456	+0.34	+0.75	+0.81
Matched necked	1234	54.92 (+0.49)	61.32 (+0.59)	63.15 (+0.24)
Matched necked	3456	55.06 (+0.63)	61.47 (+0.74)	63.22 (+0.31)
Matched − classic	1234	−0.39	+0.40	+0.93
Matched − classic	3456	+0.67	+1.10	+1.08
Vanilla	1234	54.43	60.73	62.91
Vanilla	3456	—	—	63.69

Analysis decisions and evidence. An early predeclared replication target was the largest-exit necked-minus-classic WinoGrande difference. The values (+3.94 points at seed 1234, −0.07 at seed 3456) have opposite signs, locating the repeated recovery at the aggregate rather than task-specific level. The subsequent capacity-matched comparison used largest-exit seven-task accuracy as its primary endpoint, with `avg8`, byte-PPL, and individual tasks reported alongside it. Earlier drafts used descriptive decision bands of $\geq +0.7$ points for a positive result and ≤ -0.7 for a negative result. The matched-minus-classic gains are +0.84 and +0.37; the latter and their +0.60 mean fall inside the neutral band. These descriptive bands are not calibrated confidence intervals or equivalence criteria. Reporting each value and its paired comparison makes the magnitude and consistency of the observed recovery explicit.

Table 13: 3B per-task accuracy, classic-routing arms vs. their paired vanilla ladders: suite/vanilla (Δ = difference of the task means over two-decimal per-task cells — the released `aggregate.py` convention — so it can differ from arithmetic on the displayed one-decimal cells); the paper metric (`acc_norm` on arc-c/hellaswag/obqa/piqa, raw acc on arc-e/lambada/winogrande; * = raw-acc cells). Negative Δ = suite behind; the only $|\Delta| \geq 3$ entries are the 3B-exit winogrande cells ($\approx 2.0\sigma$ and $\approx 3.3\sigma$, paired evaluation-sampling intervals).

Smol2-TH-3B-untied Task (floor)	500M	1.5B	3B
arc_easy (25)	58.5/59.5 (−1.0)	66.2/66.3 (−0.1)	69.1/68.9 (+0.2)
arc_challenge (25)	28.2/28.2 (+0.1)	33.9/32.8 (+1.1)	34.8/35.8 (−0.9)
hellaswag (25)	44.9/44.1 (+0.8)	56.0/55.6 (+0.4)	59.4/60.4 (−1.0)
lambada_openai (≈0)	44.5/44.2 (+0.3)	52.7/53.0 (−0.3)	55.7/56.9 (−1.2)
openbookqa (25)	33.2/32.8 (+0.4)	34.8/35.2 (−0.4)	37.8/37.6 (+0.2)
piqa (50)	69.6/69.5 (+0.2)	73.1/72.7 (+0.4)	74.7/74.4 (+0.3)
winogrande (50)	53.2/50.7 (+2.5)	55.9/57.7 (−1.8)	54.8/58.8 (−4.0)*
avg	**47.44/46.98** (+0.46)	**53.22/53.33** (−0.11)	**55.17/56.11** (−0.94)
G4-TH-3B-tied Task (floor)	500M	1.5B	3B
arc_easy (25)	58.0/58.1 (−0.1)	65.2/65.4 (−0.2)	68.1/69.5 (−1.5)
arc_challenge (25)	26.8/27.1 (−0.3)	32.5/31.2 (+1.3)	34.3/36.8 (−2.5)
hellaswag (25)	46.2/44.7 (+1.5)	55.7/56.4 (−0.6)	59.4/61.7 (−2.3)
lambada_openai (≈0)	49.2/48.0 (+1.2)	58.5/57.8 (+0.7)	60.9/61.2 (−0.3)
openbookqa (25)	33.0/33.0 (+0.0)	35.4/35.6 (−0.2)	38.2/37.8 (+0.4)
piqa (50)	70.2/70.2 (+0.1)	73.2/73.1 (+0.1)	74.3/74.2 (+0.1)
winogrande (50)	52.4/52.4 (+0.0)*	57.1/58.2 (−1.1)*	55.7/62.4 (−6.6)*
avg	**47.97/47.64** (+0.33)	**53.95/53.97** (−0.02)	**55.84/57.65** (−1.81)

* winogrande uses raw acc (no `acc_norm` defined).

E Winogrande mechanism probes

The per-document artifact characterizes the confidence of the checkpoints' differing predictions. Among smollm2 exit-2 errors made by the suite but not its vanilla, the suite's top1–top2 log-likelihood margin averages 0.53 nats, versus 1.05 for its remaining errors (0.86 over all suite errors). WinoGrande twin-pair identifiers are absent from the retained export, so earlier pair-correlation analyses are excluded from the reported evidence. The retained margins provide a reproducible, descriptive view of the errors that distinguish these checkpoints.

F 3B cooldown probes

Three arms characterize late-stage optimization by branching from the smol suite's pre-anneal checkpoint (iteration 30,000) and running its remaining 3.4k-iteration cooldown (Table 23). A 2×-LR reshape of the widest-exit parameters (`s3x2`) changes byte-PPL by at most 0.0001 at every exit. Decaying the minimum LR to zero (`minlr0`) changes average accuracy by at

Table 14: 3B per-task accuracy, the two necked arms vs. their paired vanilla ladders — same conventions as table 13 (the paper metric; * = raw acc). At seed 1234 the smol `Smol2-TNH-3B-untied` redesign exceeds its ladder on the aggregate at every exit despite mixed task cells and lifts winogrande to parity (−0.1); the seed-3456 pair (table 17) shows the classic arm itself at −0.3 there, so the task-level pattern is seed-specific. The g4 `G4-TNH-3B-tied` redesign holds the sub-exits but keeps the 3B signature (winogrande −4.6, arc_easy −1.1).

Smol2-TNH-3B-untied Task (floor)	500M	1.5B	3B
arc_easy (25)	58.6/59.5 (−0.9)	65.2/66.3 (−1.1)	69.4/68.9 (+0.5)
arc_challenge (25)	28.5/28.2 (+0.3)	32.4/32.8 (−0.3)	36.7/35.8 (+0.9)
hellaswag (25)	44.7/44.1 (+0.5)	56.4/55.6 (+0.8)	60.1/60.4 (−0.2)
lambada_openai (≈0)	43.6/44.2 (−0.5)	54.8/53.0 (+1.8)	56.6/56.9 (−0.3)
openbookqa (25)	32.4/32.8 (−0.4)	37.0/35.2 (+1.8)	38.0/37.6 (+0.4)
piqa (50)	70.3/69.5 (+0.9)	73.6/72.7 (+0.9)	74.5/74.4 (+0.1)
winogrande (50)	54.9/50.7 (+4.2)*	58.5/57.7 (+0.8)*	58.7/58.8 (−0.1)*
avg	**47.57/46.98** (+0.59)	**53.99/53.33** (+0.66)	**56.29/56.11** (+0.18)
G4-TNH-3B-tied Task (floor)	500M	1.5B	3B
arc_easy (25)	58.3/58.1 (+0.2)	65.7/65.4 (+0.3)	68.4/69.5 (−1.1)
arc_challenge (25)	26.3/27.1 (−0.9)	34.0/31.2 (+2.7)	34.2/36.8 (−2.6)
hellaswag (25)	46.0/44.7 (+1.4)	57.0/56.4 (+0.6)	60.0/61.7 (−1.7)
lambada_openai (≈0)	49.9/48.0 (+1.9)	59.9/57.8 (+2.1)	61.4/61.2 (+0.2)
openbookqa (25)	30.8/33.0 (−2.2)	34.8/35.6 (−0.8)	36.0/37.8 (−1.8)
piqa (50)	69.9/70.2 (−0.3)	72.5/73.1 (−0.6)	74.9/74.2 (+0.7)
winogrande (50)	51.6/52.4 (−0.8)*	57.4/58.2 (−0.9)*	57.8/62.4 (−4.6)*
avg	**47.55/47.64** (−0.09)	**54.47/53.97** (+0.50)	**56.11/57.65** (−1.54)

* winogrande uses raw acc (no `acc_norm` defined).

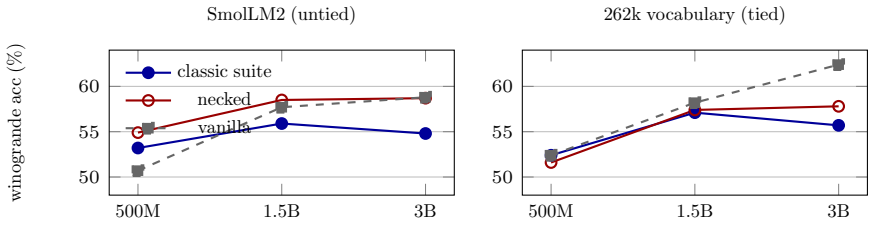


Figure 3: The winogrande stall with depth (raw accuracy; 1,267-item validation set). Left: SmolLM2 family; right: 262k-vocabulary tied family. Both classic suites stall or dip from exit-1 to exit-2 while their vanilla ladders climb (dashed); the necked redesigns (open markers; exit-private neck layers with early junction taps, section 3.1) recover part of the gap on both arms — to within 0.1 pt of vanilla at exit-2 on smol, +2.1 pts over the classic with a −4.6 pt residual on the 262k tied arm (table 12). All curves are seed 1234; at seed 3456 the smol classic arm reaches 58.5 at exit-2 (table 17), so the task-level stall is specific to seed 1234. The curves expose this seed dependence alongside the paired aggregate comparisons.

most 0.05 points and byte-PPL by at most 0.0001. Adding a ctx-gated junction-delta variant during cooldown (`jconcat-anneal`) lowers accuracy by 1.6–1.8 pts, most at the largest exit. All three continuations retain the WinoGrande gap. The probes assess late-stage sensitivity from a common checkpoint, complementing the full-training comparisons of the architectural redesign.

Table 15: Capacity-matched 3B necked sibling vs. the smollm2 vanilla ladder at both training seeds, with every paper-panel task and exit reported. Values are suite/vanilla; conventions match table 13. The last two rows of each block report the aggregate and the independently defined `avg8`; its additional raw-acc inputs, including SciQ, appear in table 22.

Task (floor)	500M	1.5B	3B
arc_easy (25)	58.4/59.5 (−1.1)	67.0/66.3 (+0.7)	70.1/68.9 (+1.2)
arc_challenge (25)	28.0/28.2 (−0.2)	33.6/32.8 (+0.9)	35.3/35.8 (−0.4)
hellaswag (25)	44.2/44.1 (+0.1)	56.6/55.6 (+1.0)	60.8/60.4 (+0.4)
lambada_openai (≈0)	43.2/44.2 (−0.9)	54.4/53.0 (+1.3)	56.7/56.9 (−0.2)
openbookqa (25)	32.2/32.8 (−0.6)	36.0/35.2 (+0.8)	38.6/37.6 (+1.0)
piqa (50)	68.7/69.5 (−0.8)	72.9/72.7 (+0.2)	74.6/74.4 (+0.2)
winogrande (50)	53.6/50.7 (+2.9)*	56.0/57.7 (−1.7)*	56.0/58.8 (−2.8)*
avg	**46.91/46.98** (−0.07)	**53.78/53.33** (+0.45)	**56.01/56.11** (−0.10)
AVG8	**54.92/54.43** (+0.49)	**61.32/60.73** (+0.59)	**63.15/62.91** (+0.24)
Smol2-TNH-3B-matched, seed 3456			
Task (floor)	500M	1.5B	3B
arc_easy (25)	57.9/59.5 (−1.6)	66.1/66.3 (−0.2)	69.0/68.9 (+0.2)
arc_challenge (25)	27.8/28.2 (−0.3)	31.7/32.8 (−1.0)	36.4/35.8 (+0.6)
hellaswag (25)	45.5/44.1 (+1.3)	56.3/55.6 (+0.7)	60.6/60.4 (+0.2)
lambada_openai (≈0)	45.3/44.2 (+1.1)	53.6/53.0 (+0.6)	56.1/56.9 (−0.8)
openbookqa (25)	32.8/32.8 (+0.0)	36.4/35.2 (+1.2)	37.4/37.6 (−0.2)
piqa (50)	68.9/69.5 (−0.5)	73.2/72.7 (+0.5)	75.0/74.4 (+0.5)
winogrande (50)	53.2/50.7 (+2.5)*	57.8/57.7 (+0.1)*	57.3/58.8 (−1.5)*
avg	**47.33/46.98** (+0.35)	**53.58/53.33** (+0.25)	**55.96/56.11** (−0.15)
AVG8	**55.06/54.43** (+0.63)	**61.47/60.73** (+0.74)	**63.22/62.91** (+0.31)

* winogrande uses raw acc (no `acc_norm` defined).

Table 16: The ctx-gate junction arm (`Smol2-TNH-3B-jcat`; junction concatenation with context gating) vs. the smollm2 vanilla ladder — same conventions as table 13. This block archives the evaluation behind the ctx-gate sentence in section 5: the arm tracks the ladder at the sub-exits and remains $-0.88/-0.43$ on the two panels at exit-2.

Task (floor)	500M	1.5B	3B
arc_easy (25)	57.5/59.5 (−2.0)	66.3/66.3 (+0.0)	68.2/68.9 (−0.6)
arc_challenge (25)	27.4/28.2 (−0.8)	33.4/32.8 (+0.6)	34.7/35.8 (−1.0)
hellaswag (25)	44.8/44.1 (+0.7)	56.4/55.6 (+0.8)	59.4/60.4 (−1.0)
lambada_openai (≈0)	44.2/44.2 (+0.1)	53.2/53.0 (+0.2)	55.8/56.9 (−1.1)
openbookqa (25)	32.8/32.8 (+0.0)	35.6/35.2 (+0.4)	36.6/37.6 (−1.0)
piqa (50)	69.4/69.5 (−0.1)	72.9/72.7 (+0.2)	74.5/74.4 (+0.1)
winogrande (50)	53.7/50.7 (+3.0)*	57.8/57.7 (+0.1)*	57.4/58.8 (−1.4)*
avg	**47.11/46.98** (+0.13)	**53.64/53.33** (+0.31)	**55.23/56.11** (−0.88)
AVG8	**54.41/54.43** (−0.02)	**61.05/60.73** (+0.32)	**62.48/62.91** (−0.43)

* winogrande uses raw acc (no `acc_norm` defined).

G Tied exit-2 diagnostics: head-block rescaling and gradient exposure

Three exploratory diagnostics characterize the 262k tied arms' exit-2 head: a finite inference-time rescaling grid, per-loss gradient norms over master-table column blocks, and a dedicated-head reference. Together, they describe the trained balance and gradient exposure of these checkpoints (tables 13 and 14), providing specific observations for interpreting the tied-head configuration.

The trained balance minimizes PPL on the tested grid. The exit-2 head multiplies the master table's columns in three blocks: $A = [0{:}1024)$ (every exit), $B = [1024{:}2304)$ (exits

Table 17: Second paired training seed (3456) of the smollm2 3B classic and necked arms vs. the seed-1234 vanilla ladder (the 3B endpoint is replicated at seed 3456, table 12); conventions as in table 13. The seed-1234 pair is in tables 13 and 14. The classic arm's exit-2 WinoGrande cell is −0.3 here against −4.0 at seed 1234, while its aggregate deficit persists (−0.52); the necked arm beats the classic arm on every aggregate at every exit in both seeds.

Smol2-TH-3B-untied, seed 3456			
Task (floor)	500M	1.5B	3B
arc_easy (25)	57.4/59.5 (−2.1)	66.4/66.3 (+0.1)	69.4/68.9 (+0.5)
arc_challenge (25)	26.8/28.2 (−1.4)	32.5/32.8 (−0.2)	34.6/35.8 (−1.1)
hellaswag (25)	44.7/44.1 (+0.6)	55.4/55.6 (−0.2)	58.8/60.4 (−1.6)
lambada_openai (≈0)	42.5/44.2 (−1.6)	52.6/53.0 (−0.4)	55.7/56.9 (−1.2)
openbookqa (25)	32.6/32.8 (−0.2)	36.2/35.2 (+1.0)	37.6/37.6 (+0.0)
piqa (50)	68.3/69.5 (−1.2)	72.1/72.7 (−0.5)	74.5/74.4 (+0.1)
winogrande (50)	52.2/50.7 (+1.6)*	55.8/57.7 (−1.9)*	58.5/58.8 (−0.3)*
avg	**46.36/46.98** (−0.62)	**53.00/53.33** (−0.33)	**55.59/56.11** (−0.52)
AVG8	**54.39/54.43** (−0.04)	**60.37/60.73** (−0.36)	**62.14/62.91** (−0.77)
Smol2-TNH-3B-untied, seed 3456			
Task (floor)	500M	1.5B	3B
arc_easy (25)	58.4/59.5 (−1.0)	65.8/66.3 (−0.5)	69.3/68.9 (+0.4)
arc_challenge (25)	26.6/28.2 (−1.5)	34.0/32.8 (+1.3)	35.7/35.8 (−0.1)
hellaswag (25)	44.3/44.1 (+0.2)	55.7/55.6 (+0.1)	59.9/60.4 (−0.5)
lambada_openai (≈0)	43.4/44.2 (−0.8)	53.3/53.0 (+0.2)	55.7/56.9 (−1.2)
openbookqa (25)	33.4/32.8 (+0.6)	37.8/35.2 (+2.6)	38.2/37.6 (+0.6)
piqa (50)	69.3/69.5 (−0.2)	73.7/72.7 (+1.0)	75.0/74.4 (+0.6)
winogrande (50)	50.8/50.7 (+0.1)*	59.1/57.7 (+1.4)*	58.4/58.8 (−0.4)*
avg	**46.59/46.98** (−0.39)	**54.20/53.33** (+0.87)	**56.03/56.11** (−0.08)
AVG8	**54.73/54.43** (+0.30)	**61.12/60.73** (+0.39)	**62.95/62.91** (+0.04)

* winogrande uses raw acc (no `acc_norm` defined).

1–2), $C = [2304{:}4352)$ (exit-2 only). Scaling the A- and B-column contributions to every logit by (α_a, α_b), C fixed, re-weights the trained balance with zero training; the trunk forward is α-independent, so the whole 7×7 grid over $\alpha \in \{0.5, 0.7, 0.85, 1.0, 1.15, 1.3, 1.5\}$ costs one forward pass per corpus. Table 24: the trained balance $(1, 1)$ is the exact grid minimum on both OOD PPL corpora. Other grid points increase PPL, reaching 2.64× at $(0.5, 0.5)$ on WikiText-103 and 3.49× at $(0.5, 1.5)$ on PG19. The block-partials path reproduces the single-GEMM baseline to machine precision. Within this finite linear grid (table 25), the trained weighting achieves the best measured PPL on both corpora; learned, nonlinear, and task-specific calibration remain separate interventions.

Gradient exposure differs across column blocks. For each suite loss term (`ce_exit0`–`ce_exit2`, `distill_exit0`–`distill_exit1`) we backward through a fresh forward on held-out wikitext-103 (12 batches × 2,048 tokens) and record the gradient L2 norm the term delivers to the master table per column block, separating the head path (the term's head GEMM, read off a leaf clone of the master) from the input path (the master as input embedding and junction-delta input). Table 26: the three CE terms have comparable head-path norms on block A, and exit-2 CE is the largest single term for `G4-TNH-3B-tied`. Under the actual coefficients (0.7 for exit-0/1 CE, 1.0 for exit-2 CE, and 0.3 for each student distillation term), the weighted sums of *individual norms* are $A/B/C = 3.354/0.709/0.450$ for `G4-TNH-3B-tied` and 3.170/0.767/0.370 for `G4-TH-3B-tied`. Normalizing those proxies by block width gives A-to-C ratios of 14.9 and 17.1. These sums describe individual-term exposure; vector alignment and cancellation in the optimizer's combined gradient are unobserved. The measurement therefore quantifies exposure differences rather than gradient starvation or a causal mechanism for accuracy. Individual head-path norms are generally 3–8 times their input-path counterparts, and the two distillation terms contribute about 8–9% of the weighted A-block norm sum.

Table 18: 3B per-task accuracy for the payback (`G4-TNH-3B-tied-payback`) and untied (`G4-TH-3B-untied`) 262k-vocabulary arms vs. their paired vanilla ladders — same conventions as table 13 (the paper metric; * = raw acc).

G4-TNH-3B-tied-payback Task (floor)	500M	1.5B	3B
arc_easy (25)	58.4/58.1 (+0.3)	65.2/65.4 (−0.2)	69.6/69.5 (+0.1)
arc_challenge (25)	29.0/27.1 (+1.9)	31.9/31.2 (+0.7)	34.7/36.8 (−2.0)
hellaswag (25)	46.8/44.7 (+2.1)	56.3/56.4 (−0.1)	60.7/61.7 (−1.0)
lambada_openai (≈0)	50.6/48.0 (+2.6)	59.6/57.8 (+1.7)	61.5/61.2 (+0.3)
openbookqa (25)	33.8/33.0 (+0.8)	35.8/35.6 (+0.2)	37.6/37.8 (−0.2)
piqa (50)	69.4/70.2 (−0.8)	72.4/73.1 (−0.7)	74.4/74.2 (+0.2)
winogrande (50)	53.9/52.4 (+1.5)*	56.9/58.2 (−1.3)*	59.1/62.4 (−3.2)*
avg	**48.85/47.64** (+1.21)	**54.01/53.97** (+0.04)	**56.81/57.65** (−0.84)
G4-TH-3B-untied Task (floor)	500M	1.5B	3B
arc_easy (25)	58.8/58.1 (+0.7)	64.9/65.4 (−0.5)	67.5/69.5 (−2.0)
arc_challenge (25)	27.3/27.1 (+0.2)	32.5/31.2 (+1.3)	35.5/36.8 (−1.3)
hellaswag (25)	45.4/44.7 (+0.8)	55.9/56.4 (−0.5)	58.5/61.7 (−3.2)
lambada_openai (≈0)	48.7/48.0 (+0.7)	57.8/57.8 (−0.1)	59.4/61.2 (−1.8)
openbookqa (25)	33.4/33.0 (+0.4)	35.8/35.6 (+0.2)	37.0/37.8 (−0.8)
piqa (50)	70.2/70.2 (+0.0)	73.2/73.1 (+0.2)	74.5/74.2 (+0.3)
winogrande (50)	53.8/52.4 (+1.4)*	57.9/58.2 (−0.4)*	58.3/62.4 (−4.1)*
avg	**48.23/47.64** (+0.59)	**53.99/53.97** (+0.02)	**55.82/57.65** (−1.83)

* winogrande uses raw acc (no `acc_norm` defined).

Dedicated-head reference. The 262k vanilla 3B's dedicated [262,144 × 2,560] head, trained by one unit-weight CE term, has per-column gradient norms of $7.9/8.2/8.2/10.7\times10^{-4}$ across width quarters. The tied exit-2's exclusive C columns receive 1.8–2.2×10^{-4} per column from `ce_exit2`. Width, hidden states, loss composition, and architecture differ between these models, so the comparison is descriptive. It supplies a standalone reference for the magnitude of exposure measured in the tied head.

Interpretation. The trained head weighting achieves the best PPL within the tested linear grid, and the per-term norms identify uneven exposure across master-table blocks. These are checkpoint-level observations: causal attribution of the accuracy gap and evaluation of other calibration functions require separate interventions. Together, the diagnostics identify concrete loss-weight and head-design variables for controlled training comparisons.

H Training-loss figures

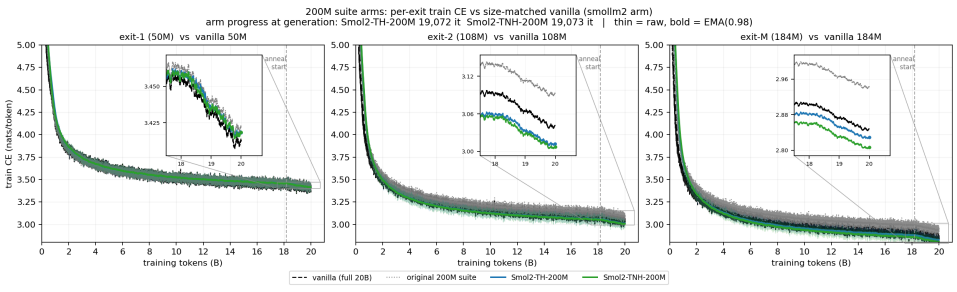


Figure 4: Suite-200M arms vs. the vanilla ladder, training loss by exit (smollm2 arm, 20B tokens). The `Smol2-TH-200M`/`Smol2-TNH-200M` arms track or beat the vanilla curve at every exit; the retired classic-capacity suite (gray) separates at the larger exits.

Table 19: 3B OOD byte-PPL per corpus (the paper's reported metric): suite/vanilla ($\Delta\%$ = (suite − vanilla)/vanilla; negative = suite better). Blocks: `Smol2-TH-3B-untied`, `Smol2-TNH-3B-untied`, its capacity-matched sibling at both seeds, and the seed-3456 classic and necked pair; the four g4 arms — the tied redesigns (`G4-TH-3B-tied`, `G4-TNH-3B-tied`), trunk-payback (`G4-TNH-3B-tied-payback`), and untied `G4-TH-3B-untied` — continue in table 20. **avg** = arithmetic mean over the 4 corpora. The same evaluations under token-PPL preserve every within-arm verdict, with relative deltas ≈4× larger (token-PPL is tokenizer-specific and not comparable across arms).

Corpus	500M	1.5B	3B
Smol2-TH-3B-untied			
wikitext-103	1.8550/1.8608 (−0.31%)	1.7513/1.7474 (+0.22%)	1.7066/1.7056 (+0.06%)
pg19	2.2009/2.2097 (−0.40%)	2.0780/2.0745 (+0.17%)	2.0281/2.0186 (+0.47%)
arxiv	1.8937/1.8965 (−0.15%)	1.8018/1.7986 (+0.18%)	1.7656/1.7622 (+0.19%)
pubmed_central	1.7366/1.7408 (−0.24%)	1.6535/1.6522 (+0.08%)	1.6249/1.6214 (+0.21%)
avg	**1.9216/1.9269** (−0.28%)	**1.8212/1.8182** (+0.16%)	**1.7813/1.7770** (+0.24%)
Smol2-TNH-3B-untied			
wikitext-103	1.8484/1.8608 (−0.67%)	1.7423/1.7474 (−0.29%)	1.6997/1.7056 (−0.35%)
pg19	2.1972/2.2097 (−0.56%)	2.0648/2.0745 (−0.47%)	2.0151/2.0186 (−0.18%)
arxiv	1.8880/1.8965 (−0.45%)	1.7919/1.7986 (−0.37%)	1.7567/1.7622 (−0.31%)
pubmed_central	1.7314/1.7408 (−0.54%)	1.6480/1.6522 (−0.25%)	1.6170/1.6214 (−0.27%)
avg	**1.9163/1.9269** (−0.55%)	**1.8117/1.8182** (−0.35%)	**1.7721/1.7770** (−0.27%)
Smol2-TNH-3B-matched			
wikitext-103	1.8537/1.8608 (−0.38%)	1.7436/1.7474 (−0.22%)	1.7014/1.7056 (−0.25%)
pg19	2.2058/2.2097 (−0.18%)	2.0662/2.0745 (−0.40%)	2.0126/2.0186 (−0.30%)
arxiv	1.8950/1.8965 (−0.08%)	1.7933/1.7986 (−0.29%)	1.7553/1.7622 (−0.39%)
pubmed_central	1.7371/1.7408 (−0.22%)	1.6475/1.6522 (−0.29%)	1.6158/1.6214 (−0.35%)
avg	**1.9229/1.9269** (−0.21%)	**1.8126/1.8182** (−0.31%)	**1.7713/1.7770** (−0.32%)
Smol2-TNH-3B-matched, seed 3456			
wikitext-103	1.8479/1.8608 (−0.69%)	1.7406/1.7474 (−0.39%)	1.6993/1.7056 (−0.37%)
pg19	2.1992/2.2097 (−0.47%)	2.0644/2.0745 (−0.49%)	2.0147/2.0186 (−0.19%)
arxiv	1.8903/1.8965 (−0.32%)	1.7908/1.7986 (−0.44%)	1.7573/1.7622 (−0.28%)
pubmed_central	1.7316/1.7408 (−0.53%)	1.6456/1.6522 (−0.40%)	1.6171/1.6214 (−0.26%)
avg	**1.9173/1.9269** (−0.50%)	**1.8103/1.8182** (−0.43%)	**1.7721/1.7770** (−0.27%)
Smol2-TH-3B-untied, seed 3456			
wikitext-103	1.8567/1.8608 (−0.22%)	1.7541/1.7474 (+0.38%)	1.7103/1.7056 (+0.27%)
pg19	2.2029/2.2097 (−0.31%)	2.0791/2.0745 (+0.22%)	2.0295/2.0186 (+0.54%)
arxiv	1.8950/1.8965 (−0.08%)	1.8049/1.7986 (+0.35%)	1.7709/1.7622 (+0.49%)
pubmed_central	1.7373/1.7408 (−0.20%)	1.6562/1.6522 (+0.24%)	1.6253/1.6214 (+0.24%)
avg	**1.9230/1.9269** (−0.21%)	**1.8236/1.8182** (+0.30%)	**1.7840/1.7770** (+0.40%)
Smol2-TNH-3B-untied, seed 3456			
wikitext-103	1.8531/1.8608 (−0.41%)	1.7434/1.7474 (−0.23%)	1.7002/1.7056 (−0.32%)
pg19	2.1984/2.2097 (−0.51%)	2.0665/2.0745 (−0.39%)	2.0120/2.0186 (−0.33%)
arxiv	1.8886/1.8965 (−0.41%)	1.7897/1.7986 (−0.49%)	1.7547/1.7622 (−0.42%)
pubmed_central	1.7344/1.7408 (−0.37%)	1.6486/1.6522 (−0.21%)	1.6151/1.6214 (−0.39%)
avg	**1.9187/1.9269** (−0.43%)	**1.8121/1.8182** (−0.34%)	**1.7705/1.7770** (−0.36%)

I Batch-scaling ratios

The timing protocol makes process-level variation explicit. A single-start `G4-TH-3B-tied` exit-0 measurement gives 36.1 ms per batch-1 step versus 26–27 ms for the identical geometry in sibling runs, identifying a warm-up artifact; exit-0 prefill and the other cells agree with those siblings. The audited blocks of table 27 randomize backend-block, boot, and exit order across three fresh processes and pool the measurements. Per-restart exit-0 medians vary by up to 0.8 ms (FA2), 1.6 ms (cuDNN fused), and 3.2 ms (eager), substantially less than that

Table 20: 3B OOD byte-PPL per corpus, continued: the 262k-vocabulary arms (same conventions as table 19).

Corpus	500M	1.5B	3B
G4-TH-3B-tied			
wikitext-103	1.8500/1.8578 (−0.42%)	1.7531/1.7541 (−0.06%)	1.7132/1.7078 (+0.32%)
pg19	2.2320/2.2309 (+0.05%)	2.1091/2.1305 (−1.00%)	2.0625/2.0737 (−0.54%)
arxiv	1.9351/1.9402 (−0.26%)	1.8409/1.8427 (−0.10%)	1.8024/1.7977 (+0.26%)
pubmed_central	1.7477/1.7590 (−0.64%)	1.6701/1.6716 (−0.09%)	1.6406/1.6353 (+0.33%)
avg	**1.9412/1.9470** (−0.30%)	**1.8433/1.8497** (−0.35%)	**1.8047/1.8036** (+0.06%)
G4-TNH-3B-tied			
wikitext-103	1.8466/1.8578 (−0.60%)	1.7437/1.7541 (−0.59%)	1.7049/1.7078 (−0.18%)
pg19	2.2167/2.2309 (−0.64%)	2.1139/2.1305 (−0.78%)	2.0507/2.0737 (−1.11%)
arxiv	1.9322/1.9402 (−0.41%)	1.8321/1.8427 (−0.58%)	1.7980/1.7977 (+0.02%)
pubmed_central	1.7427/1.7590 (−0.93%)	1.6636/1.6716 (−0.48%)	1.6318/1.6353 (−0.21%)
avg	**1.9345/1.9470** (−0.64%)	**1.8383/1.8497** (−0.62%)	**1.7963/1.8036** (−0.40%)
G4-TNH-3B-tied-payback			
wikitext-103	1.8399/1.8578 (−0.96%)	1.7449/1.7541 (−0.52%)	1.7037/1.7078 (−0.24%)
pg19	2.2176/2.2309 (−0.60%)	2.1087/2.1305 (−1.02%)	2.0598/2.0737 (−0.67%)
arxiv	1.9230/1.9402 (−0.88%)	1.8296/1.8427 (−0.71%)	1.7900/1.7977 (−0.43%)
pubmed_central	1.7379/1.7590 (−1.20%)	1.6624/1.6716 (−0.55%)	1.6302/1.6353 (−0.31%)
avg	**1.9296/1.9470** (−0.89%)	**1.8364/1.8497** (−0.72%)	**1.7959/1.8036** (−0.43%)
G4-TH-3B-untied			
wikitext-103	1.8555/1.8578 (−0.13%)	1.7546/1.7541 (+0.03%)	1.7182/1.7078 (+0.60%)
pg19	2.2827/2.2309 (+2.32%)	2.1368/2.1305 (+0.30%)	2.0990/2.0737 (+1.22%)
arxiv	1.9375/1.9402 (−0.14%)	1.8483/1.8427 (+0.30%)	1.8105/1.7977 (+0.71%)
pubmed_central	1.7559/1.7590 (−0.18%)	1.6734/1.6716 (+0.11%)	1.6452/1.6353 (+0.61%)
avg	**1.9579/1.9470** (+0.56%)	**1.8533/1.8497** (+0.19%)	**1.8182/1.8036** (+0.81%)

Table 21: Exit-2 PG19 robustness audit on all 50 validation books, separate from the fixed 20-book PPL panel. Concatenated-stream losses are attributed to books by scored-token span and converted to bits per byte (BPB). $\Delta = A - B$; intervals are 95% paired document-bootstrap intervals (10,000 resamples, seed 0), and W/L counts compare per-book BPB (lower is better). These intervals condition on the trained checkpoints and do not represent training-seed variation.

Pair $A - B$	mean A/B	Δ BPB	95% CI	W/L
matched − classic	0.9613/0.9723	−0.0109	[−0.0115, −0.0103]	50/0
necked − classic	0.9632/0.9723	−0.0091	[−0.0097, −0.0085]	50/0
classic − vanilla	0.9723/0.9666	+0.0056	[+0.0046, +0.0069]	1/49
necked − vanilla	0.9632/0.9666	−0.0035	[−0.0043, −0.0026]	49/1
matched − vanilla	0.9613/0.9666	−0.0053	[−0.0061, −0.0044]	50/0

initial artifact. One eager vanilla-3B restart is $\sim 30\%$ slower in both metrics; pooling retains it, while the sibling restarts agree with the fused and FA2 references. The single-start FA2 prefill ratios in table 29 agree with the audited medians within $\leq 5\%$. These recorded checks make both the repeated-measurement result and the exploratory batch sweep traceable to their timing protocols.

J Offline answer-deferral analysis: one checkpoint, several exits

The shared checkpoint supports a second deployment analysis: selecting among its exits using their predictions. We evaluate answer deferral offline from saved outputs (fig. 7 and table 31);

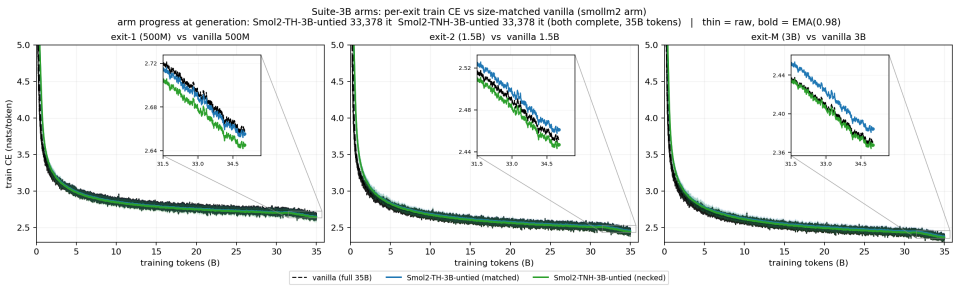


Figure 5: Smollm2 arm, Suite-3B (35B tokens): `Smol2-TH-3B-untied` (classic) and `Smol2-TNH-3B-untied` (necked) training loss by exit vs. the vanilla ladder; thin = raw, bold = EMA(0.98), inset = final 3.5B tokens.

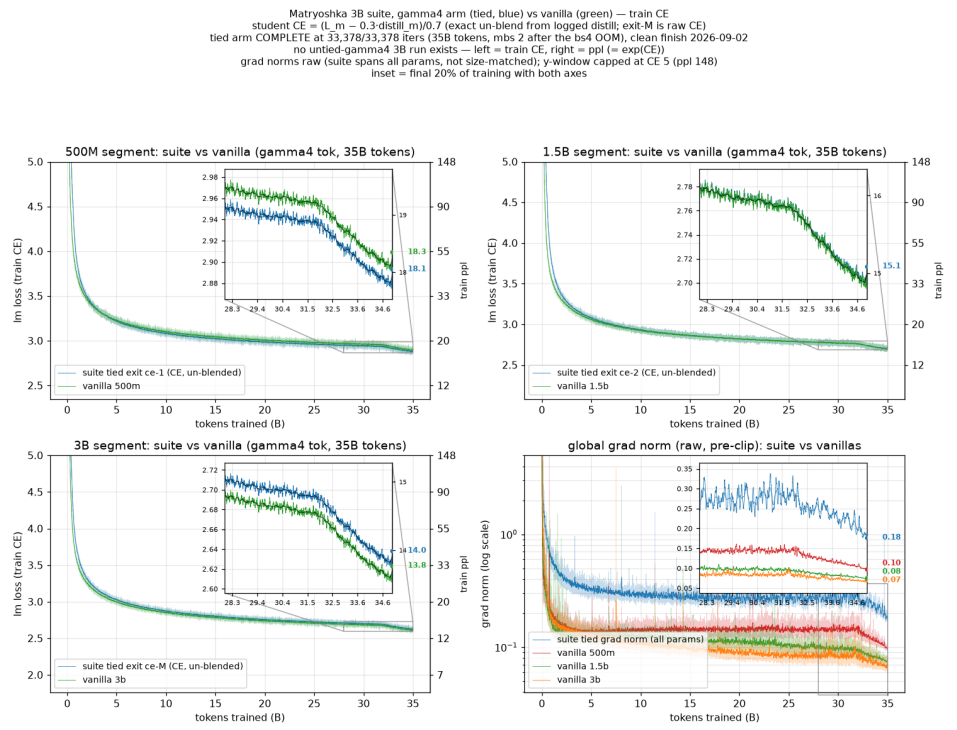


Figure 6: 262k-vocabulary arm, Suite-3B (35B tokens): the tied redesign runs' training-loss traces (per-exit losses and the shared LM loss) alongside their vanilla references.

conditional execution and its latency remain unmeasured. The resulting curves map prediction-level quality to escalation choices, complementing the fixed-exit runtime measurements.

One checkpoint exposes several inference paths. A trained suite yields, from one checkpoint, a set of standalone models of different sizes (at 3B necked: 478M/1.37B/3.1B in use; the tied 262k-vocabulary redesigns: 646M/1.67B/3.46B, with the single master table serving as every exit's LM head as well) — the smaller members with no dependence on the larger ones. This is the same single-artifact advantage as matryoshka embeddings (Kusupati et al., 2022), extended to whole language models. Deployment claims should state which residency regime they price: a *static export* holds one exit's resident tensors; a *selected-exit-hot* server holds one exit's tensors plus the shared state it executes; an *all-exits-hot* server holds the union — what *suite storage* measures (for tied arms, the complete master table); and a *standalone* baseline holds one independently trained model. Our per-exit rows price the static-export regime; *NE+LH* quantifies logical per-token mass, with resident memory accounted

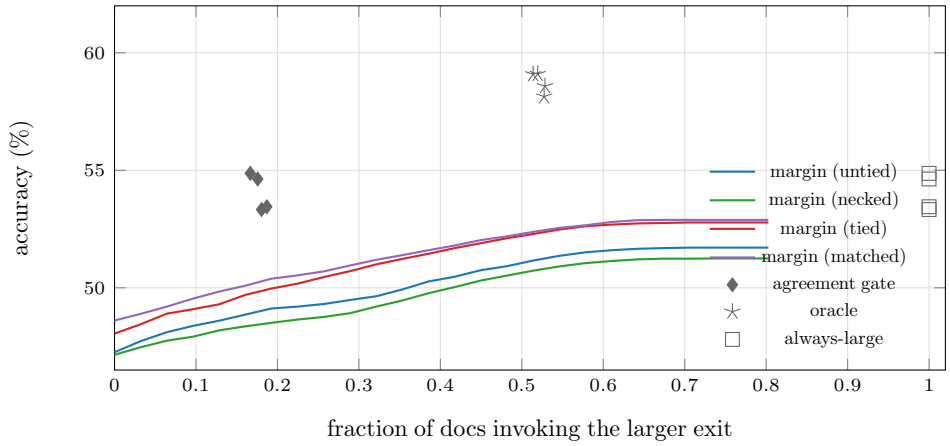


Figure 7: Cascade accuracy–escalation tradeoff, one pair (exits 0→1; all three pairs in table 31): doc-weighted pooled `avg8` accuracy (`avg8-pool`) vs. the fraction of docs whose *answer* comes from the larger exit, four 3B suites (the margin curve is an offline compute-deferral proxy; the agreement diamond is answer-deferral and requires both exits). Solid: the small exit's confidence-margin router at every budget (domain ends at ∼80%; single-resp docs stay small). Diamonds: the agreement gate — always-large accuracy at the 16.7–18.7% disagreement rate of this pair. Stars: oracle. Open squares: always-large.

for separately. Our batch-1 measurements characterize fixed exits on H100. Runtime exit selection, weight residency, batching, and synchronization are additional serving variables identified by Kumar et al. (2026); our fixed-exit measurements provide the per-model reference for such systems-level comparisons. Appendix J.1 prices the artifact advantage against the ladder at the same 35B-token budget: the suite trains for at most the ladder's matmul FLOPs (≤950 vs. 976 EF, upper-bound accounting) and its single checkpoint directory stores less than the ladder's three (243 vs. 275 GB). Seven-task accuracy is higher at each exit in this seed-1234 comparison; the paired results give its panel- and seed-specific context (section 5). Batch-1 FA2 prefill is approximately matched at exit-0 and slower at the later exits. Table 32 presents the artifact, quality, and timing measurements together for this deployment choice.

KV reuse follows the actually shared body. Classic exits are exact layer prefixes, so an exit-m KV cache is a byte-identical prefix of exit-m''s over the full shallow path. Necked exits are branches, not exact prefixes: only their shared-body layers can be reused, and private-neck entries must be discarded. At 200M necked the all-exit shared body is 5 of 21 exit-path layers, and the exit-1/exit-2 pair shares 11 of 21; at 3B necked the exit-0/exit-1 pair shares 22 of 32 layers and exit-1/exit-2 shares 30 of 36. Because the private layers are the widest, the reusable fraction of the larger exit's KV *bytes* — the quantity a transfer moves — is smaller than the layer fraction: 49.4% for exit-0→1 and 61.1% for exit-1→2 (staged KV geometry of section 6). These are geometric reuse fractions; transfer and cross-device continuation latency remain unmeasured. Prior edge–cloud work explores separately trained models (Zhu and Yang, 2025; Venkatesha et al., 2025), and LayerSkip uses same-device prefix reuse (Elhoushi et al., 2024). The shared-body accounting specifies the reusable state for an implementation of our branched suites.

Offline escalation analysis. For this analysis, we record predictions from every exit on the same evaluation items and select among exits of one network (Chen et al., 2024; Ding et al., 2024). We re-scored the zero-shot panel of all four 3B suites — `Smol2-TH-3B-untied`, `Smol2-TNH-3B-untied` (the necked redesign), and the tied redesigns `G4-TH-3B-tied` and `G4-TNH-3B-tied` — recording per-doc predictions at every exit, joined the exits' docs by (subtask, doc id), and evaluated three routers on the `avg8` panel re-pooled doc-weighted (`avg8-pool`; table 31; 26,118 joined docs per suite). Despite their shared body layers, the exits make different errors: an oracle that escalates exactly the small exit's errors reaches

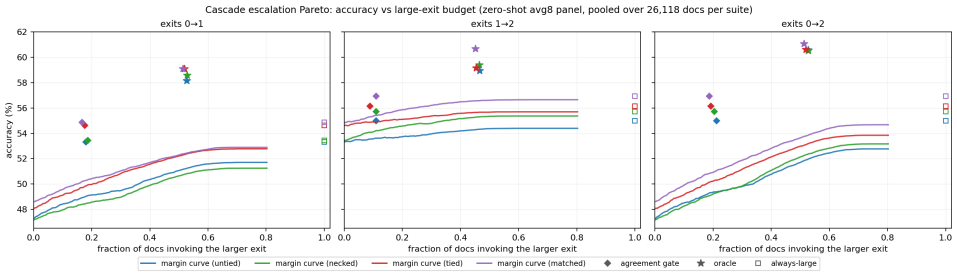


Figure 8: Cascade escalation Pareto: doc-weighted pooled `avg8` (`avg8-pool`) accuracy vs. the fraction of docs whose *answer* comes from the larger exit, for the four 3B suites of table 31 (the agreement diamond is answer-deferral: both exits execute; the margin curve is an offline proxy for compute deferral, not a timed execution). Solid curves: the small exit's confidence-margin router at every budget; its domain ends at ∼80% (single-resp docs carry no margin and stay small). Unfilled squares: always-large. Diamonds: the agreement gate — always-large accuracy at the 8.8–21.3% disagreement rate. Stars: the oracle.

58.1/58.9 at exits 0→1/1→2, +4.8/+3.9 over always running the larger exit, while touching it on only 52.7/46.7% of docs, and the surplus is also present in the other tokenizer family: +4.5/+3.0 on `G4-TH-3B-tied` and +4.2/+3.8 on `G4-TNH-3B-tied`. The evaluated signals recover part of this headroom. Prediction *agreement* is the sharper gate: on the 8.8–21.3% of docs where two exits disagree, deferring to the larger exit alone reproduces always-large accuracy at every pair — by construction, since agreeing docs share one prediction — so the gate buys always-large quality at a 79–91% small-exit *answer* rate. *Execution requirements of the selection rules.* The agreement gate is an analytical answer-deferral ceiling because it executes both exits. The confidence-margin rule can decide from the small exit alone, but we evaluate it only by replaying stored predictions. Conditional execution overhead, calibration drift, cache continuation, and end-to-end latency remain unmeasured. The small exit's own confidence (top1−top2 loglikelihood margin) needs no large-exit compute on unescalated docs: at a 25% escalation budget it buys +2.0/+0.5 of the +6.0/+1.7-point small-to-large gap (+2.3 of +6.5 on `G4-TH-3B-tied`, +2.1 of +6.3 on `G4-TNH-3B-tied`), and at 50% it buys +3.8/+1.1; pooled margins mix tasks with different choice counts, and within task the gate is sharper (on `Smol2-TH-3B-untied`, boolq *exceeds* always-large quality at the 25% budget: 62.6 vs. 61.5). Both gates are compatible in principle with the reuse property above: classic arms retain the full small-exit prefix, whereas necked arms retain only shared-body entries. Figure 8 plots the full tradeoff: the margin curve's domain ends at the ∼80% of docs that carry a margin (single-resp docs stay small by design). At equal larger-exit *answer fractions*, agreement lies above the margin curve, with the oracle above both. Agreement requires both predictions, so these are answer-fraction rather than compute-matched comparisons. The distinction makes each rule's execution requirements explicit alongside its observed accuracy.

Deployment interpretation. Capacity and hardware specify the comparison. The original necked exit-2 has +3.7% parameters relative to vanilla, so its quality and timing are interpreted alongside the classic control, the matched sibling, and the tied redesigns' NE+LH accounting. Our measurements cover single-GPU H100 fixed-exit execution. On-device NPUs, cross-device transfer, and conditional serving involve additional costs that remain unmeasured here. Related edge–cloud work (Apple, 2024; Zhu and Yang, 2025; Venkatesha et al., 2025; Lyu et al., 2026) places these costs in a broader serving context: Lyu et al. (2026) analyze WAN round-trip time and multi-tenant throughput as determinants of when speculation pays. The fixed-exit measurements and shared-body accounting provide defined inputs for evaluating those wider deployment settings.

Table 22: 3B-only extra tasks (acc, the standard metric for each task; mmlu is the macro average over its subject subtasks), suite/vanilla (Δ), added to the panel on the 3B arm. Chance floor in parentheses: mmlu/sciq 25%, boolq 50%, commonsense_qa 20%. MMLU stays near its chance floor (diagnostic only, section 5); boolq splits by arm, and commonsense_qa and SciQ are mixed. SciQ reports raw accuracy here, the value used by `avg8`, rather than `acc_norm`.

	Smol2-TH-3B-untied		
Task (floor)	500M	1.5B	3B
mmlu (25)	23.6/24.1 (−0.5)	24.8/25.4 (−0.5)	23.4/25.7 (−2.3)
boolq (50)	60.5/56.6 (+3.9)	61.5/60.0 (+1.6)	59.6/58.4 (+1.3)
commonsense_qa (20)	19.9/21.7 (−1.8)	21.1/20.3 (+0.8)	20.5/22.0 (−1.6)
sciq (25)	83.1/82.8 (+0.3)	88.1/87.8 (+0.3)	89.7/89.8 (−0.1)
	Smol2-TNH-3B-untied		
Task (floor)	500M	1.5B	3B
mmlu (25)	25.9/24.1 (+1.8)	25.2/25.4 (−0.2)	24.7/25.7 (−0.9)
boolq (50)	59.6/56.6 (+3.0)	59.4/60.0 (−0.6)	61.4/58.4 (+3.1)
commonsense_qa (20)	19.1/21.7 (−2.6)	19.2/20.3 (−1.1)	20.2/22.0 (−1.8)
sciq (25)	83.1/82.8 (+0.3)	88.3/87.8 (+0.5)	90.1/89.8 (+0.3)
	Smol2-TNH-3B-matched		
Task (floor)	500M	1.5B	3B
mmlu (25)	23.7/24.1 (−0.3)	25.1/25.4 (−0.3)	24.5/25.7 (−1.1)
boolq (50)	59.3/56.6 (+2.7)	61.6/60.0 (+1.6)	62.1/58.4 (+3.8)
commonsense_qa (20)	19.2/21.7 (−2.4)	20.7/20.3 (+0.5)	18.0/22.0 (−4.0)
sciq (25)	83.9/82.8 (+1.1)	88.5/87.8 (+0.7)	89.6/89.8 (−0.2)
	Smol2-TNH-3B-matched, seed 3456		
Task (floor)	500M	1.5B	3B
mmlu (25)	26.9/24.1 (+2.8)	27.2/25.4 (+1.8)	25.3/25.7 (−0.3)
boolq (50)	58.9/56.6 (+2.3)	63.0/60.0 (+3.1)	60.5/58.4 (+2.1)
commonsense_qa (20)	21.2/21.7 (−0.5)	21.0/20.3 (+0.7)	21.1/22.0 (−1.0)
sciq (25)	83.1/82.8 (+0.3)	90.0/87.8 (+2.2)	90.9/89.8 (+1.1)
	Smol2-TH-3B-untied, seed 3456		
Task (floor)	500M	1.5B	3B
mmlu (25)	24.7/24.1 (+0.6)	24.6/25.4 (−0.8)	26.3/25.7 (+0.6)
boolq (50)	60.8/56.6 (+4.1)	60.5/60.0 (+0.5)	56.1/58.4 (−2.3)
commonsense_qa (20)	20.2/21.7 (−1.5)	21.5/20.3 (+1.2)	19.2/22.0 (−2.8)
sciq (25)	82.4/82.8 (−0.4)	87.7/87.8 (−0.1)	89.5/89.8 (−0.3)
	Smol2-TNH-3B-untied, seed 3456		
Task (floor)	500M	1.5B	3B
mmlu (25)	24.5/24.1 (+0.4)	24.5/25.4 (−0.9)	24.6/25.7 (−1.0)
boolq (50)	61.5/56.6 (+4.9)	57.6/60.0 (−2.4)	59.5/58.4 (+1.2)
commonsense_qa (20)	19.7/21.7 (−2.0)	20.6/20.3 (+0.3)	20.4/22.0 (−1.6)
sciq (25)	83.6/82.8 (+0.8)	89.7/87.8 (+1.9)	90.1/89.8 (+0.3)
	G4-TH-3B-tied		
Task (floor)	500M	1.5B	3B
mmlu (25)	24.0/25.1 (−1.0)	25.0/24.7 (+0.2)	24.6/24.1 (+0.6)
boolq (50)	56.0/57.1 (−1.1)	62.6/62.8 (−0.2)	62.8/63.6 (−0.8)
commonsense_qa (20)	19.3/20.6 (−1.2)	20.0/19.1 (+0.9)	17.9/19.9 (−2.0)
sciq (25)	86.2/83.9 (+2.3)	88.5/89.2 (−0.7)	89.8/91.1 (−1.3)
	G4-TNH-3B-tied		
Task (floor)	500M	1.5B	3B
mmlu (25)	24.6/25.1 (−0.5)	24.4/24.7 (−0.3)	25.5/24.1 (+1.4)
boolq (50)	58.2/57.1 (+1.1)	60.8/62.8 (−2.0)	63.9/63.6 (+0.3)
commonsense_qa (20)	19.5/20.6 (−1.1)	19.9/19.1 (+0.8)	20.0/19.9 (+0.1)
sciq (25)	86.0/83.9 (+2.1)	88.4/89.2 (−0.8)	90.2/91.1 (−0.9)

Table 23: 3B cooldown arms vs. references (paper-metric avg acc; OOD byte-PPL, arithmetic mean over the four corpora — the same aggregation as table 19; parentheses: arm − suite, unrounded (may differ from displayed-cell arithmetic by ±0.01). `minlr0` re-runs the cooldown with the minimum LR decayed to zero.

	suite	vanilla	`s3x2`	`jconcat`	`minlr0`
avg acc (7 tasks, %)					
500M	47.44	46.98	47.53 (+0.09)	45.80 (−1.64)	47.42 (−0.02)
1.5B	53.22	53.33	53.25 (+0.03)	51.60 (−1.61)	53.22 (+0.01)
3B	55.17	56.11	55.14 (−0.03)	53.41 (−1.77)	55.22 (+0.05)
OOD byte-PPL (arith. mean, lower is better)					
500M	1.9216	1.9269	1.9216 (+0.0000)	1.9453 (+0.0237)	1.9215 (−0.0000)
1.5B	1.8212	1.8182	1.8211 (−0.0000)	1.8456 (+0.0244)	1.8212 (−0.0000)
3B	1.7813	1.7770	1.7813 (+0.0000)	1.8062 (+0.0250)	1.7813 (+0.0001)

Table 24: Exit-2 head block-rescale sweep on `G4-TNH-3B-tied` (logit $= \alpha_a\, \mathrm{p_A} + \alpha_b\, \mathrm{p_B} + \mathrm{p_C}$ over master-table column blocks A/B, C at 1), token-PPL on the OOD PPL corpora. Best alternative = the best of the 48 non-baseline grid points; worst = the grid maximum.

Corpus	$(1,1)$ PPL	best alternative	Δ	worst (α_a, α_b)
wikitext-103	11.49	11.68 $(1.00, 0.85)$	+1.6%	30.35 $(0.50, 0.50)$
pg19	17.62	18.00 $(1.00, 0.85)$	+2.1%	61.56 $(0.50, 1.50)$

Table 25: Full rescale grid, wikitext-103 token-PPL (rows α_a, columns α_b). Bold: the trained balance $(1, 1)$, the grid minimum.

	α_b=0.5	0.7	0.85	1.0	1.15	1.3	1.5
$\alpha_a = 0.5$	30.35	24.18	22.38	22.09	22.98	24.99	29.60
$\alpha_a = 0.7$	18.00	15.02	14.13	13.98	14.45	15.49	17.86
$\alpha_a = 0.85$	15.02	12.83	12.14	12.00	12.32	13.06	14.75
$\alpha_a = 1.0$	14.29	12.34	11.68	**11.49**	11.70	12.26	13.59
$\alpha_a = 1.15$	15.02	12.98	12.22	11.93	12.01	12.43	13.51
$\alpha_a = 1.3$	17.03	14.63	13.65	13.17	13.09	13.36	14.23
$\alpha_a = 1.5$	21.97	18.59	17.08	16.20	15.80	15.80	16.37

Table 26: Per-term, per-block gradient norms of the tied master table (wikitext-103, 12 batches × 2,048 tokens). $\mathrm{A}_h/\mathrm{A}_i$: head-path / input-path grad L2 norm into block A; likewise B, C. —: the term does not reach the block by construction.

	`G4-TNH-3B-tied`						`G4-TH-3B-tied`					
Term	A_h	A_i	B_h	B_i	C_h	C_i	A_h	A_i	B_h	B_i	C_h	C_i
`ce_exit0`	1.31	0.29	—	—	—	—	1.29	0.28	—	—	—	—
`ce_exit1`	0.88	0.22	0.47	0.14	—	—	0.98	0.21	0.45	0.15	—	—
`ce_exit2`	1.53	0.20	0.32	0.10	0.45	0.11	1.32	0.19	0.41	0.12	0.37	0.11
`distill_exit0`	0.58	0.15	—	—	—	—	0.55	0.13	—	—	—	—
`distill_exit1`	0.39	0.07	0.20	0.07	—	—	0.32	0.05	0.14	0.05	—	—

Table 27: Inference-cost parity, 3B suite exits vs. paired vanillas, as suite/vanilla ratios: parameters (Total), per-token matmul FLOPs at S=2,048, and measured single-GPU latency (H100-80GB, TP=1, batch 1, bf16). Prefill = one forward over 2,048 tokens; step = one generated token. Top three blocks use the audited protocol — three randomized restarts (backend block, boot order, and exit order randomized; distinct seeds) × 30 timed iterations after 5 warm-up, pooled medians (n=90 per cell; per-restart exit-0 medians scatter by ≤3.2 ms — see the notes above): the classic (`Smol2-TH-3B-untied`) and necked (`Smol2-TNH-3B-untied`) arms, and a capacity-matched necked variant (exits 0–1 parameter-identical to the necked arm; exit-2 re-dosed to +1.1% of vanilla; trained sibling whose quality is in tables 12 and 15). Bottom two blocks (the tied 262k-vocabulary redesigns) retain the earlier single-start protocol (median of 20). Backends: `unfused` = eager, `fused` = cuDNN fused, `FA2` = flash-attention 2. Suite exits run the deployed early-exit path (trunk prefix + exit norm/head). The tied g4 arms' params column is the tied accounting — the suite stores no separate LM head, so exit-0 reads 0.706 on Total at *identical* NE+LH (646M both sides) — and the FLOPs column confirms the per-token cost tracks NE+LH, not stored Total. [b]warm-up artifact of that single-start measurement, not a geometry effect (see the measurement notes above).

			prefill 2048			step, 1 token		
Pair	params	FLOPs	eager	fused	FA2	eager	fused	FA2
classic arm (`Smol2-TH-3B-untied`; audited)								
exit-0 (24L) / van-500M (24L)	1.000	1.000	1.00	1.00	1.01	0.98	1.00	1.02
exit-1 (34L) / van-1.5B (28L)	0.983	0.964	1.05	1.18	1.23	1.22	1.23	1.24
exit-2 (39L) / van-3B (28L)	1.006	0.975	1.72	1.52	1.54	1.41	1.43	1.43
necked arm (`Smol2-TNH-3B-untied`; audited)								
exit-0 (24L) / van-500M (24L)	1.000	1.000	1.01	1.00	1.00	0.99	1.00	1.01
exit-1 (32L) / van-1.5B (28L)	0.961	0.938	1.07	1.18	1.22	1.26	1.23	1.23
exit-2 (36L) / van-3B (28L)	1.037	1.001	1.84	1.59	1.60	1.47	1.45	1.46
capacity-matched necked (audited)								
exit-0 (24L) / van-500M (24L)	1.000	1.000	0.99	1.00	1.00	0.98	1.00	1.01
exit-1 (32L) / van-1.5B (28L)	0.961	0.938	1.05	1.19	1.22	1.22	1.24	1.23
exit-2 (35L) / van-3B (28L)	1.011	0.973	1.84	1.57	1.60	1.45	1.46	1.45
`G4-TH-3B-tied` (neck-free NE+LH re-budget, seg 24/9/4; single-start protocol)								
exit-0 (24L) / van-500M (24L)	0.706	1.000	1.03	0.99	1.04	1.02	1.05	1.38[b]
exit-1 (33L) / van-1.5B (28L)	0.786	0.989	1.09	1.23	1.25	1.19	1.26	1.16
exit-2 (37L) / van-3B (28L)	0.823	0.976	1.69	1.40	1.50	1.47	1.34	1.38
`G4-TNH-3B-tied` (necked at NE+LH parity; single-start protocol)								
exit-0 (24L) / van-500M (24L)	0.706	1.000	1.04	1.00	1.03	1.08	1.12	0.94
exit-1 (31L) / van-1.5B (28L)	0.772	0.968	1.08	1.27	1.24	1.14	1.25	1.22
exit-2 (34L) / van-3B (28L)	0.846	0.999	1.68	1.53	1.51	1.51	1.42	1.39

Table 28: Absolute batch-1 latencies behind the audited ratios of table 27: H100-80GB HBM3 (79.1 GB usable, 132 SMs), TP=1, bf16, PyTorch 2.10.0a0 (NGC 25.12), CUDA 13.1, cuDNN 9.17; three fresh processes (seeds 20260920–22) with randomized backend-block, boot and exit order, 5 warm-up + 30 timed iterations per cell, CUDA-event timing. Prefill = one forward over 2,048 tokens; step = one generated token. Each cell: pooled median [p5, p95] over n=90 timings, then the range of the three per-restart medians. Backends: FA2 = FlashAttention-2, fused = cuDNN fused, eager = unfused. Process peak allocated memory over a whole restart (all exits of one arm) was 12.2–14.8 GB for the suite arms and 7.1–12.7 GB for the vanillas (reserved ≤16.0 GB).

		prefill 2048 (ms)		step (ms)	
Model / exit	backend	median [p5, p95]	restarts	median [p5, p95]	restarts
`Smol2-TH-3B-untied` exit-0	FA2	27.05 [26.76, 28.34]	26.8–27.1	26.30 [26.02, 26.88]	26.1–26.5
`Smol2-TH-3B-untied` exit-1	FA2	38.86 [38.44, 40.79]	38.7–39.0	37.83 [37.12, 41.16]	37.3–40.6
`Smol2-TH-3B-untied` exit-2	FA2	48.93 [48.67, 50.10]	48.7–49.1	43.47 [42.82, 43.99]	42.9–43.6
`Smol2-TNH-3B-untied` exit-0	FA2	26.94 [26.83, 30.57]	26.9–26.9	26.06 [25.85, 27.04]	25.9–26.1
`Smol2-TNH-3B-untied` exit-1	FA2	38.75 [38.54, 40.35]	38.7–38.8	37.35 [37.21, 38.12]	37.3–37.5
`Smol2-TNH-3B-untied` exit-2	FA2	50.84 [50.51, 52.76]	50.6–51.4	44.41 [43.97, 45.92]	44.1–44.5
`Smol2-TNH-3B-matched` exit-0	FA2	27.00 [26.76, 27.94]	26.9–27.3	26.21 [25.83, 26.68]	25.9–26.4
`Smol2-TNH-3B-matched` exit-1	FA2	38.55 [38.10, 40.90]	38.2–39.2	37.53 [36.91, 40.25]	37.0–37.9
`Smol2-TNH-3B-matched` exit-2	FA2	50.71 [50.40, 55.04]	50.6–54.5	44.06 [43.72, 46.48]	43.8–44.8
vanilla 500M	FA2	26.90 [26.62, 27.42]	26.7–26.9	25.87 [25.68, 27.78]	25.7–26.6
vanilla 1.5B	FA2	31.70 [31.30, 32.89]	31.4–31.7	30.40 [29.93, 30.78]	30.0–30.6
vanilla 3B	FA2	31.74 [31.31, 35.89]	31.4–31.9	30.38 [30.16, 33.96]	30.2–30.6
`Smol2-TH-3B-untied` exit-0	fused	28.17 [27.92, 28.86]	28.0–28.7	27.08 [26.79, 28.63]	26.9–27.1
`Smol2-TH-3B-untied` exit-1	fused	40.31 [40.05, 41.66]	40.2–40.4	38.72 [38.36, 40.10]	38.5–38.9
`Smol2-TH-3B-untied` exit-2	fused	50.16 [49.85, 51.30]	50.0–50.4	45.07 [44.46, 46.81]	44.6–45.2
`Smol2-TNH-3B-untied` exit-0	fused	27.99 [27.74, 31.13]	27.8–29.4	27.04 [26.59, 28.64]	26.6–27.8
`Smol2-TNH-3B-untied` exit-1	fused	40.19 [39.72, 42.45]	39.8–42.0	38.62 [38.11, 41.20]	38.2–40.7
`Smol2-TNH-3B-untied` exit-2	fused	52.38 [51.54, 54.54]	51.9–52.5	45.88 [45.24, 46.78]	45.3–46.5
`Smol2-TNH-3B-matched` exit-0	fused	28.11 [27.82, 29.41]	27.9–28.5	27.05 [26.65, 27.85]	26.7–27.4
`Smol2-TNH-3B-matched` exit-1	fused	40.43 [39.80, 41.05]	39.9–40.8	38.86 [38.17, 42.53]	38.3–39.4
`Smol2-TNH-3B-matched` exit-2	fused	51.74 [51.45, 53.62]	51.5–52.8	45.99 [45.39, 46.82]	45.6–46.7
vanilla 500M	fused	28.09 [27.93, 29.19]	28.0–28.9	27.04 [26.64, 27.42]	26.7–27.3
vanilla 1.5B	fused	34.06 [32.82, 36.04]	33.0–35.9	31.46 [31.10, 32.71]	31.2–31.5
vanilla 3B	fused	33.02 [32.78, 34.91]	32.9–33.3	31.60 [31.02, 32.18]	31.4–31.7
`Smol2-TH-3B-untied` exit-0	eager	32.24 [31.56, 35.12]	31.7–34.7	31.11 [30.39, 33.82]	30.5–33.7
`Smol2-TH-3B-untied` exit-1	eager	45.98 [45.28, 46.47]	45.5–46.0	44.32 [43.54, 45.59]	43.7–44.4
`Smol2-TH-3B-untied` exit-2	eager	65.24 [64.61, 66.46]	64.7–65.3	51.11 [50.48, 52.68]	50.7–51.3
`Smol2-TNH-3B-untied` exit-0	eager	32.70 [32.32, 33.56]	32.5–33.1	31.43 [31.10, 32.69]	31.2–32.1
`Smol2-TNH-3B-untied` exit-1	eager	47.10 [46.08, 48.00]	46.3–47.5	45.83 [44.72, 47.92]	44.9–46.8
`Smol2-TNH-3B-untied` exit-2	eager	69.78 [69.15, 72.47]	69.2–70.8	53.04 [52.67, 54.30]	52.8–54.0
`Smol2-TNH-3B-matched` exit-0	eager	32.04 [31.78, 33.39]	31.9–32.9	30.96 [30.75, 32.41]	30.9–32.0
`Smol2-TNH-3B-matched` exit-1	eager	45.97 [45.71, 47.02]	46.0–46.0	44.13 [43.86, 44.57]	44.0–44.3
`Smol2-TNH-3B-matched` exit-2	eager	69.44 [68.72, 71.52]	69.2–69.5	52.48 [52.12, 53.67]	52.3–52.6
vanilla 500M	eager	32.25 [31.68, 35.75]	31.8–32.9	31.61 [30.54, 36.54]	30.7–31.8
vanilla 1.5B	eager	43.92 [43.81, 44.05]	43.8–44.0	36.31 [35.89, 38.76]	36.0–36.6
vanilla 3B	eager	37.84 [37.32, 49.88]	37.4–49.7	36.19 [35.70, 48.30]	35.8–48.1

Table 29: Batch scaling, suite/vanilla latency ratios (H100-80GB, TP=1, bf16, FA2, T=2,048; earlier single-start protocol — the batch-1 prefill cells agree with the audited medians of table 27 to within ≤5%); every ratio compares the suite exit and its vanilla at the same batch size. Prefill: one forward over 2,048 tokens (last-token logits), ratio at batch 1 / 8 / 128. Cached decode: single-token steps against a 2,048-token KV cache, ratio at batch 8 / 128 ('—' = batch-128 cache plus weights does not fit in 80 GB; KV cache ≈35–38 GB). Suite-favorable FLOPs annotated per row; KV-byte ratios (suite-favorable throughout) are quoted in the text.

		prefill (B)			cached decode (B)	
Pair	FLOPs	1	8	128	8	128
classic arm						
exit-0 / van-500M	1.000	0.959	1.080	1.085	0.998	1.004
exit-1 / van-1.5B	0.964	1.210	0.960	0.946	1.237	0.972
exit-2 / van-3B	0.975	1.551	1.024	1.006	1.441	—
necked arm						
exit-0 / van-500M	1.000	0.98	1.076	1.082	1.001	1.002
exit-1 / van-1.5B	0.938	1.28	0.965	0.947	1.249	0.973
exit-2 / van-3B	1.001	1.59	1.100	1.093	1.506	—
G4-TH-3B-tied (neck-free re-budget)						
exit-0 / van-500M	1.000	1.036	1.074	1.082	1.015	1.006
exit-1 / van-1.5B	0.989	1.248	0.914	0.881	1.244	0.948
exit-2 / van-3B	0.976	1.497	0.885	0.880	1.464	—
G4-TNH-3B-tied (necked at NE+LH parity)						
exit-0 / van-500M	1.000	1.027	1.080	1.090	0.993	1.004
exit-1 / van-1.5B	0.968	1.237	0.919	0.893	1.197	0.931
exit-2 / van-3B	0.999	1.506	0.983	0.979	1.412	—

Table 30: Prefill model FLOP utilization (MFU, % of the 989 TF/s bf16 peak) backing the batch-scaling claims: measured cells from the released batch sweep (FA2, T=2,048, one session). Necked rows (†) are implied as vanilla MFU × (FLOPs ratio / latency ratio) with the ratios of table 29 (the necked sweep itself was not retained); where the classic arm has both, measured and implied agree to ≤0.1 pp. The tied-arm sweeps ran in a separate session at 59–72% MFU and are omitted; within-session latency ratios are unaffected.

Prefill run	MFU% at B8	MFU% at B128
van-500M	44.4	46.8
van-1.5B	47.3	48.1
van-3B	56.2	56.2
classic exit-0	41.2	43.2
classic exit-1	47.5	48.9
classic exit-2	53.5	54.4
necked exit-0†	41.3	43.3
necked exit-1†	46.0	47.7
necked exit-2†	51.2	51.5

Table 31: Cascade escalation over nested exits of one pretrained network (zero-shot `avg8` panel re-pooled doc-weighted, `avg8-pool`, over 26,118 joined docs per suite; accuracies in %). Four 3B suites: the SmolLM2 classic and necked arms (`Smol2-TH-3B-untied` = *untied*, `Smol2-TNH-3B-untied` = *necked*) and the tied 262k-vocabulary redesigns (`G4-TH-3B-tied` = *tied*, `G4-TNH-3B-tied` = *matched*). Cells $f \to a$: escalated fraction f and resulting accuracy a. *Oracle* escalates the small exit's errors; *agreement* escalates docs whose small- and large-exit predictions differ; *margin* escalates the small exit's 25% least-confident docs, and single-resp docs (lambada) stay on the small exit. mmlu and commonsense_qa are excluded at the chance floor. Per-doc correctness uses raw `acc` for every task (not the paper metric's `acc_norm` conventions), so absolute pooled levels sit below doc-weighted poolings of table 13; within-table comparisons are unaffected.

suite/pair	small	large	agree%	oracle	agreement	margin 25%
untied 0→1	47.3	53.3	82.0	52.7→58.1	18.0→53.3	25.0→49.3
untied 1→2	53.3	55.0	89.1	46.7→58.9	10.9→55.0	25.0→53.8
untied 0→2	47.3	55.0	78.7	52.7→60.6	21.3→55.0	25.0→49.5
necked 0→1	47.2	53.4	81.3	52.8→58.6	18.7→53.4	25.0→48.7
necked 1→2	53.4	55.7	89.0	46.6→59.4	11.0→55.7	25.0→54.8
necked 0→2	47.2	55.7	79.6	52.8→60.5	20.4→55.7	25.0→49.6
tied 0→1	48.1	54.6	82.4	51.9→59.1	17.6→54.6	25.0→50.4
tied 1→2	54.6	56.2	91.2	45.4→59.2	8.8→56.2	25.0→55.2
tied 0→2	48.1	56.2	80.7	51.9→60.6	19.3→56.2	25.0→50.7
matched 0→1	48.6	54.9	83.3	51.4→59.1	16.7→54.9	25.0→50.7
matched 1→2	54.9	56.9	89.0	45.1→60.7	11.0→56.9	25.0→56.0
matched 0→2	48.6	56.9	81.3	51.4→61.1	18.7→56.9	25.0→51.4

J.1 Resource frontier: one suite vs. the independent ladder

Table 32: Resource frontier at the token-matched 35B budget (smollm2 3B family, seed 1234): the three-exit necked suite vs. the independent vanilla ladder. Storage = size of the checkpoint directories (resumable state: model + optimizer for all exits); suite-arm directories span 178–389 GB across the paper's variants (classic to matched-vocabulary; the headline necked arm is shown). Training FLOPs = measured matmul-only per-token forward (batch sweep, S=2,048) × 3 (fwd+bwd) × 35B tokens; the suite entry upper-bounds joint training by summing the three deployed exit paths (the trunk is shared, so the true cost is lower; deepest-path-only lower bound 590 EF). Quality = the canonical avg acc, 7 tasks (table 12); latency = audited FA2 batch-1 prefill medians (table 28).

Option	params in use	storage (GB)	FLOPs (EF)	avg acc (7 tasks)	prefill FA2 (ms)
Totals					
`Smol2-TNH-3B-untied` (one run/dir)	478M–3.1B	242.9	≤950		
independent ladder (three runs/dirs)	478M–3.00B	274.8	976		
Per-size points (suite / ladder)					
exit-0 / van-500M	478M / 478M			47.57/46.98	26.94/26.90
exit-1 / van-1.5B	1.37B / 1.42B			53.99/53.33	38.75/31.70
exit-2 / van-3B	3.1B / 3.00B			56.29/56.11	50.84/31.74