CONTEXT-TOWER CONVERSION PRESERVES GENERATION WHILE FREEZING RETAINS KNOWLEDGE: LOW-BUDGET AR-TO-DIFFUSION CONVERSION OF MOE LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Converting a pretrained autoregressive (AR) model to a diffusion language model (dLLM) enables parallel generation without pretraining a new model. Published conversion methods differ by roughly three orders of magnitude in training data and have not been compared under a common protocol. We compare two conversions of the same 30B Mixture-of-Experts (MoE) parent, holding the corpus, supervised-token budget, trainable parameter set and evaluation harness fixed, each under its own training recipe. The in-place model updates a subset of the parent's weights using denoising and representation-alignment losses; the frozen-tower model instead conditions through cross-attention on a frozen causal copy of the parent. With 1 B training tokens, the frozen-tower model scores 71.60 on HumanEval pass@10 against 6.19 for the in-place model, an $11.6\times$ improvement. At the same budget it also keeps 95% of the parent's GSM8K score and 99% of its MMLU-Pro score. A dense-parent experiment reproduces the HumanEval separation. Within the two-tower design at about 500 M tokens, freezing the context tower retains substantially more MMLU-Pro performance than training it, while both give similar observed HumanEval scores. Our theoretical analysis establishes that both conversion classes contain an exact sampler for the AR parent under a hard attention mask and left-to-right commitment of one position per round. Under a shared loss, freezing removes the gradient contribution through the context states. Furthermore, evaluation protocol substantially affects a published 500 B-token conversion's scores in both directions across tasks, while its AR parent's scores vary by less than three points, so comparing dLLMs needs a common protocol. These results show that, in the tested low-budget regime, the frozen-tower configuration retains substantially more of the parent's generation performance than in-place conversion. The construction applies to any causal transformer with softmax attention.

1 INTRODUCTION

A common route to a diffusion language model (dLLM) is to initialise from a pretrained autoregressive (AR) model and continue training with a denoising objective (Gong et al., 2025; Ye et al., 2025). Existing conversion studies are difficult to compare. Their data budgets range from $\sim$1 B tokens (Wu et al., 2025) to 500 B tokens (Radical Numerics Inc., 2025), their recipes vary across several coupled design choices, and their models have not been evaluated under a common protocol.

We study low-cost AR-to-diffusion conversion under matched training and evaluation. Within the tested regime, the frozen-tower configuration, which we call context-tower conversion, retains most of the parent's generation performance at budgets where in-place conversion retains little. This gain comes from the whole configuration. A trainable-tower control gives similar observed HumanEval scores with frozen and trainable towers, while freezing retains substantially more MMLU-Pro performance (§4). We fixed the evaluation gates, numerical thresholds, and stopping criteria before each run and evaluated every model with the same fixed harness. Our contributions are:

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.

- An *output-length-dependent failure mode* for in-place conversion (§2). Across the masked-diffusion recipes, the converted models retain most of the parent's performance on knowledge benchmarks, while generation performance declines as output length increases.
- A *matched comparison of two conversion configurations* (§3). Two conversions of the same 30B MoE parent share corpus, budget, trainable parameter set and evaluation harness; one trains a subset of the parent weights under denoising and alignment losses, the other reads a frozen causal context tower (Reda et al., 2026). The frozen-tower model reaches 71.60 on HumanEval pass@10 against 6.19, a factor of $\gtrsim 11$ (§3 gives the resampled range).
- An *evaluation practice for cross-protocol comparisons* (§5). We found that decoding protocol changes a converted model's score by up to $4.8\times$ while the AR parent's scores move by less than three points. Single-draw differences below three points may disappear or reverse when decoding is repeated (§3). We therefore compare on a fixed task set with the parent as a control, report sampled and deterministic decoding separately, and re-decode with three seeds.
- A *budget-efficiency comparison*. With 1 B training tokens, the frozen-tower model reaches 74–79% of the HumanEval pass@10 score of a published 500 B-token conversion under that model's inference code.
- A *trainable-tower control* and a gradient-isolation analysis (§4). Two models trained with the same recipe to approximately 500 M tokens, one with a frozen tower and one with a trainable tower, achieve similar observed HumanEval scores, but the frozen-tower model scores 13 points higher on MMLU-Pro. Under the same loss, freezing removes one gradient term. Both configurations can reproduce the parent under sequential sampling, and Appendix B.3 runs that sampler.
- *Two ablations of the tower–denoiser interface* (§7). Attach density is varied over 25–100% of tower layers, and denoiser size over three donors at a fixed tower.

Scope. We focus on MoE parents because freezing the experts leaves only $\sim$3% of the parameters trainable. The construction applies to any causal transformer with softmax attention.

2 Output-length-dependent degradation under in-place conversion

We use OLMoE-1B-7B (Muennighoff et al., 2025), an MoE with 1 B active parameters and a public pretraining mixture. Continuing on this mixture reduces distribution shift during conversion. We test four in-place recipes: (i) full fine-tuning with masked diffusion (Sahoo et al., 2024); (ii) frozen experts with layer-wise representation alignment to the frozen AR teacher (Peng et al., 2026); (iii) block diffusion (Arriola et al., 2025; Wu et al., 2025) with an exact prefix KV cache; and (iv) uniform-state corruption with self-conditioning. Each arm uses 0.75–1 B supervised tokens.

All masked-diffusion arms show the same dependence on output length (Table 1; Appendix D measures the length axis). They retain $\approx 77\%$ of the parent's MMLU-Pro score. BoolQ remains near the parent, short extractive answers degrade, and JSON validity falls from 99.6% to 0.0–1.8%. Code generation also degrades sharply: MBPP is 0.0 for every masked-diffusion arm, and HumanEval remains far below the parent.

The uniform-state arm, evaluated with its native entropy-bound sampler, also loses single-token performance (BoolQ 12.8). That score is far below the 62% majority-class baseline. The harness counts an answer without a yes/no word as wrong, and the arm's JSON outputs show token-level corruption, so a failure to emit a valid answer token is a possible explanation. We did not count invalid BoolQ answers. Applying the same sampler to the block-model weights reduces MMLU-Pro from 15.4 to 0.6, although those weights were trained with masked diffusion. The additional loss therefore comes from the decoder, because the two columns share the same trained weights and the same training objective. The small-parent BoolQ scores are close to the $\approx 62\%$ majority-class baseline and therefore provide weak evidence for single-token retention. The 30B experiment in Figure 2 provides the stronger control.

The output-length profile is unchanged by a $25\times$ change in trainable parameters, a different attention layout, or additional training within the low-budget regime. Repeating 0.8 B tokens and training on

1 B fresh tokens produce flat trajectories at matched steps. A published 500 B-token conversion of the 30B parent (Radical Numerics Inc., 2025) reaches parity on HumanEval under its own inference code (§5), consistent with more training reducing the deficit. It does not isolate budget.

Table 1: **In-place conversion of OLMoE-1B-7B.** Each arm uses 0.75–1 B supervised tokens and the fixed evaluation harness. The lower rows report the structured-output suite (BoolQ, SQuAD-short EM and CoNLL-JSON validity). We added this suite after the phase gates and ran it on the AR base and the three arms then live. The full fine-tuning arm had been superseded by (ii) at the first gate, and the entropy-decoder column reuses (iii)'s weights and was scored on the primary suite only. [§]The same block-model weights evaluated with the entropy-bound decoder; MMLU-Pro decreases from 15.4 to 0.6. Each cell uses one sampling run; pass@10 uses $n = 20$ samples. Sampling seeds were not fixed.

	AR base	(i) full-FT	(ii) +align	(iii) block	(iii)×ent.[§]	(iv) uniform
HumanEval pass@10	22.97	1.01	1.99	1.51	0.00	0.00
GSM8K	12.59	1.67	2.12	1.82	0.15	1.59
MBPP	18.60	0.00	0.00	0.00	0.00	0.00
MMLU-Pro	19.80	15.20	15.20	15.40	0.60	8.40
BoolQ (%)	63.2	—	63.4	62.0	—	12.8
SQuAD-short EM (%)	67.4	—	44.0	22.0	—	0.6
JSON validity (%)	99.6	—	1.8	0.0	—	9.6

3 THE MATCHED PAIR

Design. INPLACE-1B and FROZEN-TOWER-1B use the same parent (Qwen3-30B-A3B-Base (Yang et al., 2025); HumanEval pass@10 89.23), corpus, supervised-token budget (1.000 B), bit-identical data manifest, trainable parameter set (0.919 B parameters), optimiser and schedule, and evaluation harness. They differ in context architecture, alignment loss, block-size schedule and corruption procedure (Appendix B): the comparison evaluates two conversion configurations and does not isolate freezing alone. FROZEN-TOWER-1B reads a frozen, layer-aligned copy of the parent through cross-attention (Figure 1); INPLACE-1B computes the same context with the weights being retrained. The tower keys, rotated at their absolute positions, and the tower values are concatenated into the denoiser's existing attention, so the attachment adds no trainable parameters.

Results (Figure 2). The in-place arm reaches HumanEval pass@10 **6.19**, within the prespecified insufficiency band (≤ 10). Its score remains flat from 250 M to 1 B tokens: $4.96 \to 4.26 \to 4.38 \to 6.19$. It also scores 0.4 on MBPP and 7.96 on GSM8K. The 30B parent therefore shows the same low-budget in-place deficit as OLMoE-1B-7B.

The frozen-tower arm reaches **71.60** on HumanEval pass@10, a $\gtrsim 11\times$ improvement and 76–80% of the parent score (seed floor to committed draw). The committed scores were obtained on each arm's training host. When both arms were evaluated on the same Blackwell host using three decoding seeds, their mean scores were 67.47 for the frozen-tower model and 4.61 for the in-place model (Appendix E). It scores 82.56 on GSM8K, 54.60 on MMLU-Pro and 53.40 on MBPP, or 95%, 99% and 72% of the parent. Its JSON validity is 96.8%, compared with 31.8% for the in-place arm and 0.0–9.6% for the small-parent in-place arms.

The in-place arm still retains 82% of the parent MMLU-Pro score (45.4 against 55.40) and does better on short and structured output than the small-parent in-place arms. Its SQuAD EM exceeds the parent score (46.0 vs. 37.6), but this difference comes from output formatting. The parent over-generates under the short-answer prompt, whereas the converted model emits length-bounded blocks. The same effect accounts for the lower SQuAD score of the 30B parent relative to the 4B parent.

Long-form generation nevertheless remains poor for the in-place arm. Because the two arms optimise the same parameter set, a difference in trainable parameter count does not explain the gap. The small-parent full-fine-tuning arm, with $25\times$ the trainable count, shows the same profile (§2). On held-out data each arm reduces its own diffusion objective by about twelve nats from initialisation (4.55 and 4.68 at 1 B tokens; Appendix B). The objectives differ so these losses do not measure relative convergence.

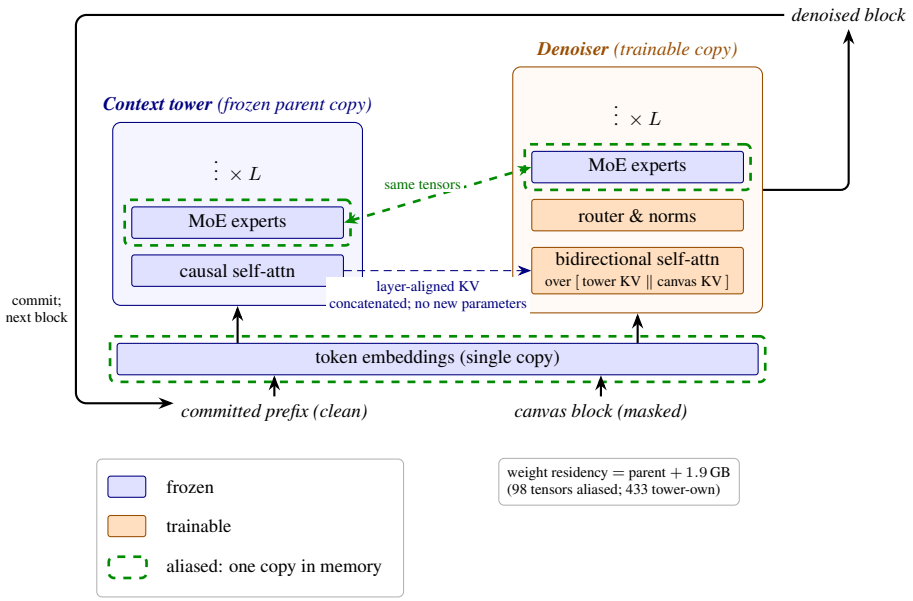


Figure 1: **The frozen-tower architecture (FROZEN-TOWER-1B).** The frozen context tower is a bit-exact parent copy running causal attention over the committed prefix; the denoiser is a second copy of the same architecture, bidirectional over the masked canvas block, reading the tower through layer-aligned cross-attention (Reda et al., 2026), implemented as key/value concatenation into the denoiser's own attention, so the attach introduces *no* parameters. One layer is expanded per tower; computation flows upward. Under the frozen-experts recipe only attention, router, and norms train (orange). Because experts and embeddings stay bit-identical to the parent, tower and denoiser *alias* them (green): the two-tower system holds parent $+\,1.9$ GB of weights at inference, under $2\times$ parent (§6).

Relation to prior two-tower work. Reda et al. (2026) introduced the frozen-tower architecture and report 98.7% quality retention at 2.42× throughput after training on 2.1 T tokens. Their Mamba-hybrid MoE parent does not permit a directly matched in-place conversion because most layers are recurrent and expose no causal attention mask to replace.

Replication across seeds and hardware. We trained a second FROZEN-TOWER-1B seed with a new data order and corruption sequence on different hardware. Two bit-parity tests found no measurable difference between the hardware implementations. At 500 M tokens, the seeds differ by less than half a percentage point on HumanEval pass@10 (63.63 vs. 64.04), within the prespecified $\pm$5-point band. Their MMLU-Pro scores are identical (54.2), and the second seed reaches 94.2% JSON validity.

At 250 M tokens, the HumanEval pass@10 scores differ by $\sim$11 percentage points and fall outside the prespecified band. This checkpoint fails the mid-training replication criterion. By 500 M tokens the difference disappears, while knowledge metrics remain within one point. For this model and training setup, mid-training HumanEval differences below $\sim$ 10 percentage points fall within the observed *training*-seed variation; MBPP and knowledge benchmarks were more stable. Resampling the decode on both arms places the 500 M comparison inside *sampling* dispersion as well: three draws give 63.86 for the first seed and 65.61 for the second, against a range of 2.24 on the second arm alone. The two seeds agree at 500 M with no resolvable ordering. We did not train a second in-place seed. The in-place profile instead appears in six arms across three parents: the four recipes in §2, INPLACE-1B on the 30B MoE, and INPLACE-DENSE on the dense parent. The 65-point pair gap is much larger than the observed frozen-tower seed variation, but does not replace replication of the in-place arm.

Sampling dispersion. All scores other than the greedy rows come from sampling, and the committed values are single draws. To see how far one draw can move, we decoded five committed cells

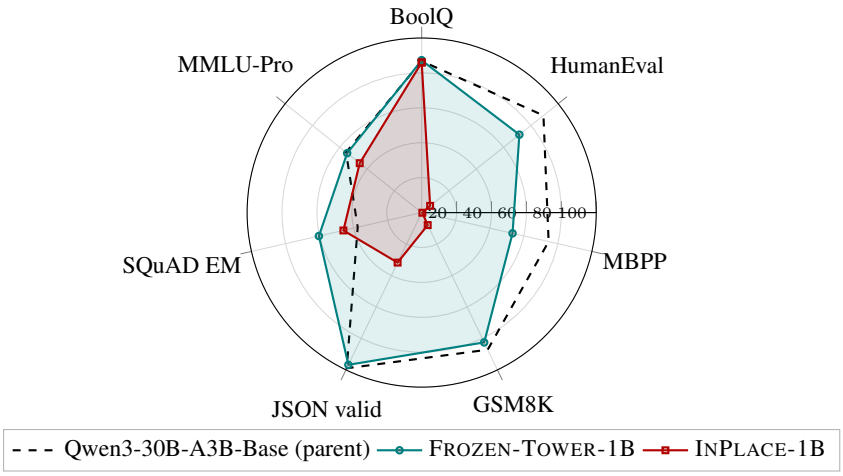


Figure 2: **The output-length-dependent degradation, at a glance** (all axes are percentages on a common 0–100 scale, frozen harness; ordered counterclockwise by output length). Both conversions stay close to the parent on the knowledge and short-output axes (BoolQ, MMLU-Pro, SQuAD), though INPLACE-1B is ten points down on MMLU-Pro and SQuAD EM sits above the parent for the formatting reason given in §3. INPLACE-1B (red) then collapses on the longer-output tasks (JSON validity 31.8, GSM8K 7.96, MBPP 0.4), while FROZEN-TOWER-1B (teal), the frozen-tower configuration, holds at or near the parent on the knowledge and structured-output axes, at 95% on GSM8K, and at 72–80% of it on the two code axes.

again with three decode seeds each, leaving the training and the checkpoints untouched. Greedy decoding gave the same score in every cell. Taken within one seed and one host, the pair ratio ranges from 12.2–18.7×. Almost all of that width comes from the in-place arm, whose three seeds span 1.86 points around a mean of 4.61. Two comparisons showed single-draw differences of less than two points. When we re-decoded both arms with three seeds, the sign of each difference flipped. So we mark single-draw differences below three points as unresolved and draw no conclusion from them. Appendix E gives the per-cell tables and the cross-host check behind the comparisons in §5 and §6.

Dense-parent replication. We repeated the matched comparison with a dense parent (Qwen3-4B-Base). We kept both configurations, the corpus, the attachment mechanism and the evaluation harness fixed and changed the parent family (Appendix C, Table 7). After 500 M supervised tokens, the frozen-tower model reaches 53.13 on HumanEval pass@10, or 61.5% of the parent score (86.42). This exceeds the prespecified retention floor of 45%. It also remains close to the parent on MMLU-Pro (45.60 vs. 46.60), BoolQ (86.60 vs. 86.80), SQuAD EM (76.00), and JSON validity (96.60%).

The matched in-place arm uses the same corpus, budget, and seed. Its MLPs and embeddings are frozen, while its attention layers and normalisation parameters are trained; this is the same trainable set as the frozen-tower denoiser. It reaches 6.55 on HumanEval pass@10 and is already flat at 250 M tokens (6.43), within the prespecified ceiling of 0.15 (ratio 0.076, or 7.6% of the parent). It scores 0.00 on MBPP and 9.80% on JSON validity while retaining 68.7% of parent MMLU-Pro. The dense comparison reproduces the length-dependent deficit at 8.1× separation on HumanEval pass@10.

Comparison with a 500 B-token conversion. RND1 uses the same parent and a closely matched freezing recipe, with a training budget of 500 B. Under its native inference code and sampled protocol, it reaches parent parity on HumanEval pass@10 (90.63 vs. 89.23), with 90.6 ± 0.1 across two disjoint seed sets. Relative to this endpoint, FROZEN-TOWER-1B reaches 74–79% of the score with 1 B supervised tokens.

The comparison uses RND1's native decoder, which is favourable to the baseline (§5). Under greedy decoding, FROZEN-TOWER-1B exceeds RND1 at every measured budget. The sampled protocol, where the baseline is strongest, is compared in §5. Its HumanEval greedy pass@1 scores are 22.56, 44.51, and 42.07 at 250 M, 500 M, and 1 B tokens, all read on one Hopper host lineage, compared

Table 2: **Trainable versus frozen context tower.** Same recipe, corpus, data order and training seed; frozen = the committed seed-1 checkpoint at 4,960 steps (about 490 M tokens), trainable at 5,059 steps (500.08 M). Differences are trainable minus frozen. †Committed-score difference; the paired re-decode comparison gives -0.33 $[-6.13, +5.26]$ (Appendix A). MMLU-Pro interval: independent-binomial approximation (Appendix A).

	HumanEval pass@10	MMLU-Pro
frozen tower	64.04	54.2
trainable tower	63.83	40.8
difference	$-0.21^{\dagger}$	-13.4 $[-19.5, -7.3]$

with 14.63 for RND1 under its native inference code. The greedy score falls by 2.4 points from 500 M to 1 B while sampled pass@10 rises from 64.04 to 71.60. Greedy has no decode dispersion (Appendix E), so the fall is a property of the checkpoint and is reported as measured.

4 THE ROLE OF FROZEN CONTEXT

The trainable-tower control. The frozen-tower recipe is rerun with the tower's attention, router and norms trainable, at the same corpus, data order and training seed (42) and approximately the same 500 M-token budget, with the bands fixed before the run (Appendix I). Table 2 gives the result. The two arms have similar observed HumanEval scores, and the paired interval includes both improvements and declines of about six points. MMLU-Pro is 13 points lower with the tower trained, an interval that excludes zero, and the gap is already present at 250 M. Training the tower also doubles the trainable count from 0.92 B to 1.84 B under the learning rate and schedule set for the smaller set (Appendix A), so part of the MMLU-Pro drop may come from optimising the larger set with those hyperparameters. This comparison does not separate that from movement of the context representation. Together with §3, these comparisons show a generation advantage for the tested two-tower configuration, with no resolved HumanEval advantage from freezing at about 500 M tokens. Freezing gives higher MMLU-Pro performance under the shared training schedule. Whether separating context computation from denoising explains the generation advantage is untested. One reading of Appendix B is that in-place training makes one set of weights do two jobs, causal context and block denoising, and the context maps move away from the parent's. The trainable tower's weights also change while its observed HumanEval score stays similar, so weight change alone does not account for the in-place loss.

What the propositions add. With parent parameters θ^*, per-layer key–value maps $\mathrm{KV}_{\theta^*}(c)$ for a committed prefix c and AR conditional π^*, the frozen-tower class is $\mathcal{G} = \{q_\phi(\cdot \mid z, \mathrm{KV}_{\theta^*}(c))\}$, with the context states fixed; the in-place class has the same form with $\mathrm{KV}_W(c)$ from the trained weights.

Proposition 1 (Parent realisability with a frozen context)**.** *Assume that attention uses a hard visibility mask, so a query at position i places zero weight on masked positions and on visible positions after i. Then $\mathcal{G}$ contains a sampler $q_{\phi^\dagger}$ with parent-copied weights and masked-key suppression. Under a left-to-right schedule that unmasks one position per round, this sampler exactly reproduces the parent's AR factorisation: $p_{\phi^\dagger}(x_B \mid c) = \prod_i \pi^*(x_i \mid c, x_{B,<i})$.*

The proof (Appendix A) is an induction over layers. The in-place class contains the same sampler at $W = \theta^*$, so realisability does not separate the arms. We built this sampler, the witness, and ran it (Appendix B.3): it agrees closely with the parent's greedy score at one position per round, and the trained checkpoint loses its score under the same decoding rules, so this existence result does not establish that the trained model preserves the parent under its native sampler. For the loss $L(\psi) = \mathbb{E}\,\ell(q_\psi(z, \mathrm{KV}_\psi(c)))$ the chain rule splits the in-place gradient into an emission term and a context term. The frozen-tower gradient is the emission term alone, and at $W = \theta^*$ the two emission terms coincide (Proposition 2, Appendix A). Freezing removes the context term. After 1 B tokens the in-place arm's context maps sit at mean relative distance 0.33 and 0.44 from the parent's (Appendix B).

Table 3: **Cross-protocol evaluation.** Two implementations on our task set, beside the published scores from the authors' harness. Parent scores agree across all three implementations, whereas RND1 varies strongly with decoding mode. *Parent results reproduced from Radical Numerics Inc. (2025). ‡Published MBPP and GSM8K results use sampled decoding on the authors' harness; the parity cells use greedy decoding on our task set. For RND1, sampled pass@1 changes from 67.10 to 14.63 under native greedy decoding. The T=0.01 rows are read in the text; on HumanEval, pass@10 collapses to pass@1 there because n=20 near-deterministic draws buy no diversity. "n/r" marks a cell the source does not report.

	published	our decode	parity (their code)
Qwen3-30B-A3B-Base MBPP / GSM8K (diagnostic)	74.2* / 85.2*	74.60 / 87.19	74.4 / 88.02
Qwen3-30B-A3B-Base HumanEval pass@10 / greedy	n/r	89.23 / 53.66	89.20 / 53.05
RND1 HumanEval pass@10 (sampled)	∼90s	18.90	**90.63**
RND1 HumanEval greedy	n/r	17.68	14.63
RND1 MBPP / GSM8K	65.4 / 80.0 (sampled)‡	18.20 / 21.53 (near-greedy)	12.00 / 9.93 (greedy)
RND1 MBPP / GSM8K / MMLU-Pro at T=0.01	n/r	18.20 / 21.53 / 43.40	19.0 / 21.0 / 41.8

5 CROSS-PROTOCOL EVALUATION

We evaluate two decoding protocols on a fixed task set and set the published scores beside them (Table 3). The AR parent moves by less than three points across the three columns. Between our decode and the baseline's own code, the largest move is 1.64 points, on HumanEval pass@1 (47.77 against 46.13), which Table 3 does not show. This three-way agreement suggests that task formatting and harness implementation are not the main source of the dLLM differences. The scores of RND1, however, move by task-dependent amounts in both directions across protocols. Under its native sampled protocol, it reaches parent parity on HumanEval (90.63; sampled pass@1 67.10), whereas our original decoder gives a score lower by a factor of 4.8. Under its native greedy mode, the score falls to 14.63 on greedy pass@1, a $4.6\times$ decrease at the same metric. It also falls below our near-greedy decoder on every deterministic task, by $2.2\times$ on GSM8K, so the disagreement runs in both directions.

This behaviour is consistent with temperature changing the unmasking order. Temperature scales the logits before the entropy-ordered unmasking schedule, so it affects both sample diversity and unmasking order. We also ran RND1 under *its own* code at temperature 0.01, which separates an implementation difference from the temperature setting itself. It scores 21.0 on GSM8K, 19.0 on MBPP and 41.8 on MMLU-Pro. These are within 1.6 points of our own 0.01 decode on all three tasks, and 5.6 to 11 points above its own greedy mode. So most of the gap on the deterministic tasks closes once the baseline's own code moves off greedy. The published sampled scores are still 3.4 to 3.8 times higher than these, but that comparison also differs in decoding protocol and evaluation harness. Cross-protocol dLLM comparisons should therefore include parent-agreement diagnostics and report sampled and deterministic decoding separately.

6 THROUGHPUT AND MEMORY

After 1 B training tokens, FROZEN-TOWER-1B commits 3.8 tokens per forward. It does not show the sharp-commit behaviour of Nemotron-Labs-TwoTower (Reda et al., 2026). A 150 M-token adaptation stage reduces the canvas block size S from 32 to 16. The first-round commit fraction increases on every task, from 21.6% to 42.5% on HumanEval, while pass@10 reads 69.66, above the prespecified floor of 65. The stage shows no detected cost to quality: resampling both arms on three decode seeds puts the adapted model at 70.40 against 68.49 for the S=32 arm, both read on Hopper hosts, a difference of 1.91 points in the adapted arm's favour with a paired interval of $[-1.9, +5.8]$ (Appendix E). The S=16 arm is the S=32 model trained for a further 150 M tokens, so the comparison mixes the smaller block with the extra budget and cannot separate them.

At batch size 1, the frozen-tower model runs at $0.134\times$ the throughput of the served AR baseline at $L = 64$ and $0.168\times$ at $L = 256$. Before the engineering pass described below, the same model ran at $0.057\times$ and $0.080\times$ on the same hardware. Both are measured on a pinned 20-prompt protocol, so the pass gained $2.34\times$ at $L = 64$ and $2.09\times$ at $L = 256$. The gain comes from an engineering pass on the canvas forward, which took 85.6% of wall-clock time at $L = 256$ beforehand. Static

Table 4: **Interface ablations at a fixed tower.** Four attach-density arms at 250 M supervised tokens with the same corpus, seed and harness. They differ in which tower layers the denoiser reads. The 25 % and 100 % arms were trained and scored on Hopper hosts, the 50 % and 75 % arms on a Blackwell host (Appendix K); seed spread and unmeasured contrasts as in the text.

attach density	HumanEval pass@10	GSM8K	MMLU-Pro	MBPP
25 % (12/48)	19.60	50.49	43.00	6.60
50 % (24/48)	40.09	66.72	57.00	35.40
75 % (36/48)	40.94	73.77	56.00	39.80
100 % (48/48)	47.11	77.33	54.60	43.60

shapes and graph capture cut that step from 91.30 ms to 16.13 ms in isolation, a factor of 5.66. With capture cost and per-call synchronisation counted, the same step gains 3.39× at $L = 64$ and 2.97× at $L = 256$. All optimised paths are bit-identical to the reference decoder. Faster variants that failed this test are excluded.

The tower and denoiser alias their frozen expert and embedding tensors. The resident weight footprint is therefore the parent model plus 1.9 GB; activations and caches are additional.

The remaining throughput deficit is not explained by weight traffic. For a 256-token completion, the converted model streams 1.32 TB of expert weights, compared with 1.69 TB for served AR. It moves 22% less weight data but remains slower. The estimated memory-bandwidth lower bound for the canvas forward is 2.5 ms, compared with the measured 16.13 ms. The profile therefore points to small-batch MoE gather and launch overhead as the main optimisation targets. We performed this optimisation after the architecture experiments, so the quality results were obtained with a fixed inference stack. All throughput numbers are at batch size 1. Larger batches may spread the MoE gather overhead over more tokens. The results establish training-budget efficiency and quality retention, but not an end-to-end serving advantage.

7 Attach density and denoiser size

The denoiser connects to the tower through two settings: which tower layers it reads, and how many parameters it has. We call these two settings the interface. In every arm so far the denoiser read every tower layer and was a full copy of the parent. This section changes one setting at a time and holds the tower, recipe, corpus, budget and harness fixed.

Attach density. Four arms read different numbers of tower layers, each trained for 250 M supervised tokens with the same data order (Table 4). The attach sets are nested and listed in Appendix F. The quarter and full arms were trained and scored on Hopper hosts and the half and three-quarter arms on a Blackwell host (Appendix K). The quarter-to-full and half-to-three-quarter comparisons are within one accelerator architecture, and comparisons between those pairs cross architectures. The one hardware effect we measured, a 1.83-point greedy shift on another checkpoint (Appendix E), does not bound hardware effects on these sampled comparisons. The first twelve added layers add 20.49 points of HumanEval pass@10, the next twelve 0.85, the last twelve 6.17. The training-seed spread of the full-density configuration at this budget is about 11 points, wider than the whole range from half to full density. We did not measure the uncertainty of the density contrasts. The metrics do not agree on which densities differ (Table 4). GSM8K keeps rising with density, so full attachment scores 10.6 points higher than half density on arithmetic, and that comparison crosses architectures. All four arms use a uniform stride, and placement at matched density is measured separately in Appendix F.

Denoiser size. Appendix F measures a smaller denoiser at a fixed tower and attach set. Two dense donors from the same family, with 0.44 B and 1.409 B trainable parameters, are compared at the same 500 M-token budget. The larger donor is worse on every metric that separates the arms. At matched tokens per trainable parameter (Appendix F) it is better on HumanEval and worse on MMLU-Pro, so a larger denoiser shows no benefit across the board. Its learning-rate path also departed from the schedule (Appendix I). The two donors also ran on different accelerator architectures with different micro-batch sizes, so this ablation is confounded three ways. We draw no size conclusion from it, and a rerun with a matched schedule, host and micro-batch is the next step.

Limitations. The dense comparison uses one seed per arm at 500 M tokens and the main MoE comparison uses 1 B tokens. The dense frozen-tower model was still improving when stopped. We replicate the frozen-tower model across seeds but not the in-place model. We did not run a frozen tower with the alignment loss; the trainable-tower control (§4) is the nearest single-variable test of freezing, at one seed and approximately one budget. The propositions address expressivity and gradient structure, the drift and held-out measurements use one checkpoint each, and none addresses optimisation dynamics or the empirical $500\times$ budget ratio. Batch-size-1 throughput remains below served AR, and all experiments use post-RoPE key/value concatenation. We vary attach density at one budget and denoiser size at two (§7), with one seed per arm. The density arms use a uniform stride, so they bound the effect of density and say nothing about placement. The committed scores are single decode draws. We measure their spread directly (§3).

8 Conclusion

Across the tested parents, the frozen-tower configuration retains substantially more generation performance than the in-place configuration at matched supervised-token budgets. Across masked-diffusion recipes on a small parent and matched MoE and dense comparisons, in-place training preserves most knowledge-benchmark performance and degrades generation on longer-output tasks. On the 30B MoE parent the frozen-tower configuration reaches 71.60 on HumanEval pass@10 against 6.19, a factor of $\gtrsim 11$. The frozen-tower model stays ahead of the in-place model at every attach density we tested. At the same token budget, a smaller denoiser sees more training tokens per trainable parameter than a larger one. A token-matched comparison therefore cannot determine whether a performance difference is caused by denoiser size or by the amount of training each parameter receives. At 1 B tokens, the frozen-tower model reaches 74–79% of the 500 B-token in-place reference score, while weight aliasing keeps resident weights at the parent plus 1.9 GB.

Within the two-tower design, two models trained with the same recipe, one with a frozen tower and one with a trainable tower, achieve similar observed code scores, but the frozen-tower model scores 13 points higher on MMLU-Pro in this comparison. The results therefore support keeping the pretrained context representation fixed and available while training the denoiser for parallel emission. The propositions show that the class contains a sampler that reproduces the parent under a sequential schedule. They also show that freezing removes one gradient term. Evaluation protocol is a separate source of variation, the same dLLM score changing by up to $4.8\times$ across the tested protocols, so comparisons need a fixed task set, parent-agreement controls and separate sampled and deterministic results.

9 Related work

A2D conversion. Masked-diffusion conversion of AR checkpoints spans DiffuGPT and DiffuLLaMA (Gong et al., 2025), Dream (Ye et al., 2025), RND1 (Radical Numerics Inc., 2025), and LLaDA2.0's 16B and 100B MoE conversions (Bie et al., 2025), at budgets from under 200 B to 500 B tokens. Fast-dLLM v2 (Wu et al., 2025) and block objectives of the same family bring the cost to $\sim$1 B tokens at 1.5–7B dense scale. FLUID (Ma et al., 2026) adapts dense models with a strictly causal alignment and an entropy-driven denoising stride at a few billion tokens. From-scratch diffusion language models at scale include LLaDA (Nie et al., 2025) and LLaDA-MoE (Zhu et al., 2025). None of this work reports a matched in-place against frozen-tower comparison on one parent, corpus, budget and trainable set, a common-protocol cross-model table, or the behaviour of expert routing under conversion. Appendix L places the objectives and samplers we build on.

Two-tower conversion. Reda et al. (2026) introduced the frozen context tower, freezing a Mamba-hybrid MoE parent and training a denoiser copy with layer-aligned cross-attention at 2.1 T tokens, retaining 98.7% of parent quality at $2.42\times$ throughput. Their parent is mostly recurrent and exposes no causal attention mask to replace, so it admits no matched in-place control. We adopt their geometry at $1/2100$ of the budget on a full-softmax parent and supply the comparison and the analysis. With weight aliasing, resident weights total the parent plus 1.9 GB, at the cost of a second forward (§6). In-place recipes such as Fast-dLLM v2 avoid the second forward.

AI USE STATEMENT

The authors used an LLM-based coding assistant for implementation and experiment operation: writing and reviewing training, evaluation and analysis code, operating runs on remote hosts, and drafting and checking prose in this paper and in the internal results documents it cites. The assistant also searched for related work and verified every reference. The propositions in §4 were formulated by the authors. The assistant checked the proofs in Appendix A and polished their writing. The assistant drafted the documents that fixed bands before each run, calibration bars and results readings from the authors' stated designs, proposed candidate experiments and readings, and produced first drafts of the analyses in the appendices. Every band, stopping rule and reading rule was approved by the authors before the corresponding run. The outcomes were not fixed in advance. Four runs departed from their documents. Appendix I records each departure and when it was written down, including one check that was adopted after its cell had been measured. Every interpretation in this paper was reviewed against the committed evidence by the authors. And every number reported here was produced by the committed code on the frozen harness and verified against the committed artefacts. The assistant is not an author and made no research contribution that the authors did not direct and check.

ETHICS STATEMENT

This work trains and evaluates language models on public benchmarks and a public pretraining mixture. It involves no human subjects and collects no data. Converted models inherit the capabilities and limitations of their parents and will be released for research use.

REPRODUCIBILITY STATEMENT

For every training run the numerical bands, stopping rules and reading rules were fixed in a signed document before the run launched, and every arm was scored on one frozen evaluation harness. Analyses added after review, such as the within-task length grouping in Appendix D, are marked as exploratory where they appear. Appendix I lists the bands the paper reads against and the four departures from them. Appendix J specifies the task suites and sampling parameters; Appendix E gives the per-seed decode tables behind §3; Appendix K inventories every arm: the configuration of record and seed, corpus, training host, and which artefacts the archived record holds. Configurations are committed once per run and never edited in place. Committed numbers are never revised; new measurements are reported as new rows, and where a later result changes an interpretation we change the wording and leave the number as measured. The configuration files, corpus manifest, evaluation specification and harness revision will be released with the code.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.09573.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, 2021a.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021b.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. LLaDA2.0: Scaling up diffusion language models to 100B. *arXiv preprint arXiv:2512.15745*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT 2019*, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast LLMs via self-distillation through time. In *ICLR*, 2025. arXiv:2410.21035.

Justin Deschenaux, Caglar Gulcehre, and Subham Sahoo. The diffusion duality, chapter II: Psi-Samplers. *arXiv preprint arXiv:2602.21185*, 2026.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *NeurIPS*, 2024. arXiv:2407.15595.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *ICML*, 2024.

Xiangyu Ma, Teng Xiao, Zuchao Li, and Lefei Zhang. From AR to diffusion: Efficiently adapting large language models with strictly causal and elastic horizons. In *Proceedings of ACL 2026*, 2026. arXiv:2605.27387.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoE: Open mixture-of-experts language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xXTkbTBmqq.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Fred Zhangzhi Peng, Alexis Fox, Anru R. Zhang, and Alexander Tong. Don't retrain, align: Adapting autoregressive LMs to diffusion LMs via representation alignment, 2026. URL https://arxiv.org/abs/2605.06885.

Radical Numerics Inc. Training diffusion language models at scale using autoregressive models. Technical report, Radical Numerics, October 2025. URL https://www.radicalnumerics.ai/assets/rnd1_report.pdf. RND1-Base-0910; code at github.com/RadicalNumerics/RND1.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP 2016*, 2016.

Fitsum Reda, John Kamalu, Roger Waleffe, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-Labs-TwoTower: Diffusion language modeling with pretrained autoregressive context, 2026. URL https://arxiv.org/abs/2606.26493.

Subham Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. In *ICML*, 2025. arXiv:2506.10892.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *ICLR*, 2025. arXiv:2412.03487.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, 2003.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 61810–61843, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dLLM v2: Efficient block-diffusion LLM, 2025. URL https://arxiv.org/abs/2509.26328.

Zirui Wu, Lin Zheng, Zhihui Xie, Jiacheng Ye, Jiahui Gao, Shansan Gong, Yansong Feng, Zhenguo Li, Wei Bi, Guorui Zhou, and Lingpeng Kong. DreamOn: Diffusion language models for code infilling beyond fixed-size canvas. *arXiv preprint arXiv:2602.01326*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tieyuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, Xiaolu Zhang, Zhenzhong Lan, Junbo Zhao, Da Zheng, Chongxuan Li, Jianguo Li, and Ji-Rong Wen. LLaDA-MoE: A sparse MoE diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025.

A PROOFS AND THE TOY MODEL

Proof of Proposition 1. Index absolute positions from 1. The committed prefix c occupies positions $1, \dots, n$ and is held by the tower. The current block occupies positions $n{+}1, \dots, n{+}S$. The canvas has $S{+}1$ slots: slot 0 is the anchor and it holds the last committed token c_n at position n, and slot $q \geq 1$ holds block position $n{+}q$. At round i ($1 \leq i \leq S$), slots $1, \dots, i{-}1$ hold the committed block tokens $x_{B,<i}$ and slots $i, \dots, S$ hold mask tokens. Define $\phi^\dagger$ by copying every module from θ^*: the experts and embeddings are bit-identical by construction, and the attention, router, normalisation layers and output head take their parent values. The visible set of a canvas query at slot q is $V_q = \{1, \dots, n\} \cup \{\text{slots } r : 1 \leq r \leq \min(q, i{-}1)\}$, the tower's keys for the prefix together with the committed canvas slots at or before the query; masked slots and later committed slots are suppressed, and the anchor's own key is suppressed because position n is already exported by the tower. The emission for block position i is read at slot $i{-}1$, the slot holding the token at position $n{+}i{-}1$ (the anchor when $i = 1$).

We show by induction over layers that, for every visible slot $r \leq i{-}1$ (the anchor included), the hidden state at slot r equals the parent's hidden state at position $n{+}r$ for the sequence $(c, x_{B,<i})$. At layer 0 the input at slot r is the embedding of the committed token at position $n{+}r$, as in the parent, and every layer rotates its queries and keys at that same absolute position. For the inductive step, assume equality through layer $l{-}1$ at every visible slot. At layer l the query at slot r attends over $[\mathrm{KV}_{\theta^*}(c) \| \mathrm{KV}(\text{slots } 1, \dots, r)]$. The tower's keys and values are the parent's for positions $1, \dots, n$, since the tower is the parent run causally; the canvas keys and values at slots $1, \dots, r$ are the parent's for positions $n{+}1, \dots, n{+}r$ by the induction hypothesis, since keys are rotated at their absolute positions and values are projected from the same hidden states. This is exactly the set the parent's causal attention sees at position $n{+}r$, and grouped-query repetition is identical, so the attention output at slot r equals the parent's at position $n{+}r$. The identical router, experts and normalisation layers close the step. The output head at slot $i{-}1$ therefore yields $\pi^*(\cdot \mid c, x_{B,<i})$, and the one-position-per-round left-to-right schedule factorises the block exactly. When the block completes, its S tokens are committed to the tower's prefix and the next block's anchor holds $x_{B,S}$ at its absolute position, so the argument repeats across block boundaries. □

The equivalence is exact under the visibility assumption. The agreement of the implemented witness with the parent in finite precision (Appendix I, row 3) is a separate check on the implementation.

Proposition 2 (Exact gradient decoupling)**.** *Let $L(\psi) = \mathbb{E}\, \ell(q_\psi(z, \mathrm{KV}_\psi(c)))$. For the in-place parameterisation, where the context states depend on the trained weights W, the gradient is $\nabla_W L = G_{\mathrm{emit}}(W) + J_W^\top G_{\mathrm{subst}}(W)$. For the frozen-tower parameterisation, the context states are constant and the gradient is G_{emit} alone; at the parent initialisation $W = \theta^*$, where the context states coincide, this term is the same in both arms. The parameterisations therefore differ by exactly one term, $J_W^\top G_{\mathrm{subst}}$, which is not constrained to preserve the witness in Proposition 1.*

Proof of Proposition 2. By the chain rule, for $L(W) = \mathbb{E}\, \ell(q_W(z, \mathrm{KV}_W(c)))$ with W appearing in both arguments, $\nabla_W L = \mathbb{E}\big[(\partial_W q)^\top \nabla_q \ell\big] + \mathbb{E}\big[(\partial_W \mathrm{KV}_W(c))^\top (\partial_{\mathrm{KV}} q)^\top \nabla_q \ell\big]$, where $\partial_W q$ is the Jacobian of q with respect to W at fixed context states, $\partial_{\mathrm{KV}} q$ is its Jacobian with respect to the context states, and $\partial_W \mathrm{KV}_W(c)$ is J_W. The first expectation is G_{emit}, and $J_W^\top G_{\mathrm{subst}}$ is shorthand for the second. The frozen-tower model computes $\mathrm{KV}_{\theta^*}(c)$ with constant θ^*, so the second term vanishes. At $W = \theta^*$, where the context states coincide, its first term is the same G_{emit} as in the in-place model. □

The two-block toy model. Consider data (x, y) with $y = A_2^* A_1^* x + \xi$, substrate $A_1 \in \mathbb{R}^{k \times d}$, emission map $A_2 \in \mathbb{R}^{m \times k}$, and squared loss. With $A_1 \equiv A_1^*$, the objective is a least-squares problem in A_2 alone. Under joint training, $\nabla_{A_1} L$ contains terms of the form $A_2^\top R x^\top$, where R is the residual. When the emission map is initially misaligned with the new task, these terms are generically nonzero, so joint training can move A_1 away from A_1^* while A_2 is still aligning. This is the mechanism Kumar et al. (2022) formalise for fine-tuning relative to linear probing; the model here is a qualitative illustration and derives no rate.

The toy model is consistent with three observations: flat low-budget in-place trajectories, insensitivity to trainable-parameter count, and a gap that increases with the required degree of parallelism. It is a qualitative illustration and does not account for the observed transformer gap.

Components of the frozen-tower intervention. The frozen-tower intervention combines three changes: a fixed context representation, an additional read path, and gradient isolation. Proposition 2 formalises the last of these. A sequence of measured interventions provides a partial test of context anchoring. On the small parent, the unanchored in-place model scores 1.01 on HumanEval pass@10. Adding layer-wise representation alignment as a loss raises the score to 1.99, with a 1.8% increase in training loss. On the 30B parent, architectural anchoring with the frozen tower reaches 71.60, compared with 6.19 for the matched in-place model. These comparisons do not isolate context anchoring: the loss-based and architectural interventions use different parent scales. The trainable-tower control below more directly tests freezing within the two-tower design and gives different results on HumanEval and MMLU-Pro. The propositions establish realisability and gradient structure, not optimisation dynamics. The $11.6\times$ pair difference and the $500\times$ budget ratio are measured rather than derived.

The trainable-tower control: record. Table 2 reads the control against the committed frozen seed-1 checkpoint. The configuration is the frozen recipe with four changes: the tower is trainable, training stops at 500 M tokens, a checkpoint is saved at step 2,480, and the tower is held resident in memory. The last of these affects where the tower's weights live and not what is computed. With the flag off, the acceptance record found identical outputs under a deterministic kernel. Corpus, data order, RNG seed 42, corruption seed, loss, block schedule, learning rate (1e-4, 2,000-step warm-up, cosine over the 1 B run) and decoding are shared.

Two differences remain. The control's final checkpoint is at 5,059 steps and 500,079,173 supervised tokens, and the frozen comparator is the lineage's step-4,960 save at about 490 M tokens, so the two sit 99 steps apart on the shared schedule. The control was also scored on an H200 and the comparators on a GH200. Both are Hopper hosts, and the decode path was checked token-identical. The second frozen seed ran 5,050 steps on 499,236,606 tokens, close to the control's budget, and scores 63.63 and 54.2. Training the tower raises the trainable count from 0.92 B to 1.84 B. Every evaluation decoded through the trained tower: 427 of the 433 tower tensors differ from the parent, and the six that do not are never read.

Uncertainty. The committed frozen HumanEval draw (64.04) has no per-problem file, so the paired intervals use the same checkpoint's three re-decodes from Appendix E. Against the re-decode at decode seed 1234 (64.15) the difference is -0.33 [-6.13, $+5.26$]. Against seeds 2345 (63.57) and 3456 (63.87) it is $+0.26$ [-5.64, $+6.41$] and -0.04 [-5.42, $+5.37$]. On HumanEval+ the three read -1.43, $+0.32$ and -0.58, with intervals of similar width. Each interval comes from 10,000 resamples of the 164 problems, with pass@10 recomputed from the per-problem counts. They describe movement across problems for these checkpoints and generations. They do not measure training-seed variability, and all three use the same control draw.

The frozen 500 M MMLU-Pro score has no per-item file, so no paired interval can be formed for that comparison. The interval in Table 2 treats the two scores as independent binomials, 204/500 against 271/500, with a standard error of 3.13 points and a normal 95 % width. It ignores the dependence that comes from scoring both models on the same questions. A paired interval exists only against the frozen 1 B final (54.6, 273/500), which gives -13.8 [-18.0, -9.4] over item resamples at a different budget.

At 250 M the control reads 41.2 against 54.6 on MMLU-Pro and 47.25 against 47.11 on HumanEval. Every other metric moves by under four points: GSM8K 77.18 against 79.76, MBPP 49.40 against 45.8, and HumanEval+ 57.10 against 57.66. The bands were 61.0–67.0 on HumanEval for a matched reading and below 51.2 on MMLU-Pro for a loss. Both were met. The registered conflict rule covers co-primaries that point in opposite directions, and a matched score points in no direction, so it was not invoked. A band is a decision rule and not an equivalence test. Training the tower costs 13.8 GiB of peak memory and makes a step $1.43\times$ slower. The control is one seed at one budget, and MMLU-Pro is one benchmark.

Scope. The propositions establish realisability and the exact one-term gradient difference between the two parameterisations. The toy model illustrates the mechanism qualitatively. These results do not predict training-budget constants for transformers; the $500\times$ ratio is measured rather than derived.

B MECHANISM MEASUREMENTS ON THE MATCHED PAIR

The two configurations of §3 share parent, corpus and data order, supervised-token budget, trainable parameter set, optimiser and schedule, denoising-objective family and harness, and differ in four things. INPLACE-1B trains under a representation-alignment loss to the frozen parent (cosine, λ=10, all layers), with a canvas block that ramps from 32 to 64 tokens at step 1000. Its corruption masks each position on an independent draw, at a rate that depends on the position. FROZEN-TOWER-1B has no alignment loss, because its tower is the parent and never moves. Its canvas block stays at 32 throughout. Its corruption masks exactly k positions per block, drawn by Gumbel top-k with a weighting towards later positions. The in-place block objective processes $4\times$ as many tokens per supervised token, so on the same hardware the frozen-tower model trains about $3.6\times$ faster per supervised token, and the 500 M-token replication run took 27.4 h on one GPU. Each architecture decodes under its own backend on one harness (Appendix J). The comparison therefore evaluates the two configurations; the trainable-tower comparison described in Appendix A tests freezing within the two-tower design, with the checkpoint-budget and host differences recorded there.

Both measurements below use the committed 1 B checkpoints of the pair and the parent, with the bands fixed before they ran. No number in §3 or §4 changed. The held-out sequences come from the shared corpus under the seed-42 permutation, from index 700,000 onward. Both runs consumed 647,040 sequences, so the held-out slice begins past everything either of them saw.

B.1 CONTEXT DRIFT UNDER IN-PLACE CONVERSION

Proposition 2 says the in-place gradient carries a term that can move $\mathrm{KV}_W(c)$ off $\mathrm{KV}_{\theta^*}(c)$. This measures whether the maps moved. On 2,000 held-out sequences, INPLACE-1B and the parent are run causally over the same clean tokens, and at every supervised position the per-layer keys (after `k_norm` and RoPE at absolute positions) and values are compared by relative Frobenius distance, with the cosine of the residual stream as a second view. The frozen arm's tower is the parent, so its distance is zero by construction. This comparison and parent-against-parent were run as sanity checks and returned exactly zero.

Reading: DRIFT-PRESENT. The prespecified band was a mean relative distance of at least 0.05 in a quarter of the layers or more. Measured: mean relative K distance 0.33 and V distance 0.44, above 0.05 in all 48 layers, with a maximum of 0.60 (layer 36, V); residual-stream cosine to the parent 0.967 on average and 0.946 at its lowest (Figure 3). The profile rises with depth, from 0.18/0.29 at layer 0 to about 0.38/0.55 over the last twelve layers: the maps moved everywhere, while the residual stream kept its direction to within a few degrees. No prediction about where drift concentrates was set down in advance, so the profile confirms none. Only the 1 B checkpoint exists (the 250 M–750 M in-place checkpoints were not archived), so no trajectory is claimed.

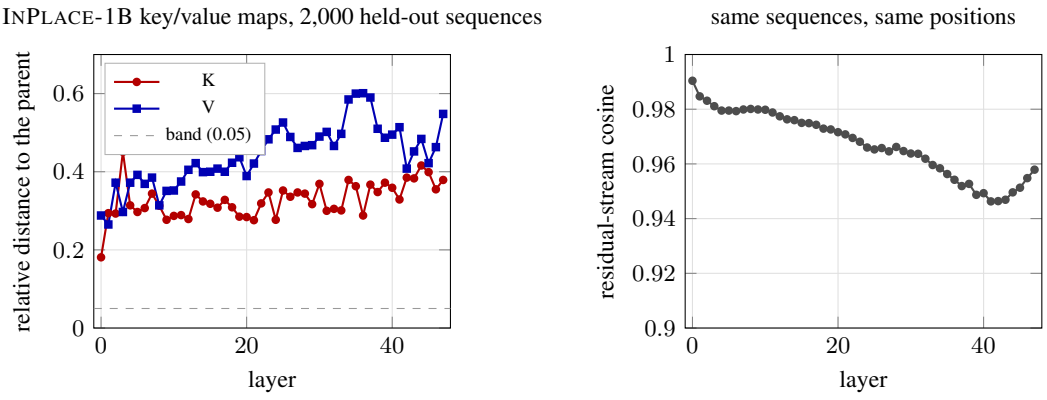


Figure 3: **Context drift of the in-place arm after 1 B tokens.** Left: per-layer relative Frobenius distance of INPLACE-1B's keys and values from the parent's on the same held-out tokens; the frozen arm sits at zero by construction. Right: cosine between the two residual streams.

B.2 HELD-OUT OBJECTIVE OF THE PAIR

We scored each arm under its own training objective, on the same 25,000 held-out sequences with the same corruption seed (777). The two kernels differ, so the arms do not see the same masks. INPLACE-1B was scored with independent per-position masking at block 64, and FROZEN-TOWER-1B with exactly-k masking at block 32. Each score is the diffusion term alone, with no representation alignment and no router auxiliaries. We scored one batch twice and got the same loss to the digit.

Reading: between the bands, reported as measured. Two bands were set in advance. OBJECTIVE-MATCHED was an in-place loss within 0.10 nats (natural-log units per scored token) of the frozen arm's, and OBJECTIVE-WORSE was a loss 0.30 nats or more above it. The measured losses are 4.547 nats for INPLACE-1B and 4.685 for FROZEN-TOWER-1B (Table 5). The in-place arm is 0.138 nats *below* the frozen arm. That falls outside the 0.10 band, on the side no band named, so the result is reported as measured.

The two losses are not on a common scale. The objectives differ in corruption kernel, mask geometry and block size, so the losses do not measure how far each arm has converged relative to the other. The OBJECTIVE-MATCHED reading would not have established that either. Both arms reduce their own denoising loss by about twelve nats from initialisation, while their generation performance differs sharply.

The last two rows of the table were added after the readings. They score the parent initialisation under each arm's objective. The arms start at 16.984 and 16.599 nats and reduce their own objectives by 12.44 and 11.91 (paired over sequences, sd 2.34 and 1.99). Each reduction is in that arm's own units and is reported as such.

Table 5: **Held-out objective of the pair.** 25,000 held-out sequences, corruption seed 777, each arm under its own objective, in nats per scored token. The parent rows score the untrained initialisation of each arm under that arm's objective. Two sequences in the frozen arm's draw carried no scored token. They enter the token-weighted mean with zero weight, and the paired statistics use the remaining 24,998. The last two rows were added after the readings.

	INPLACE-1B (in-place)	FROZEN-TOWER-1B (frozen tower)
mean loss, token-weighted	4.547	4.685
mean of per-sequence losses	4.619	4.758
paired difference, frozen − in-place	+0.139 (sd 0.581), in-place worse on 35.2% of sequences	
parent initialisation, same objective (unregistered)	16.984	16.599
improvement from initialisation	12.44	11.91

B.3 THE WITNESS AND THE K-SWEEP

Proposition 1 says the frozen-tower class contains a sampler that reproduces the parent. We call that sampler the witness, because running it shows the class really does contain one. We built it and ran it on one Hopper host (Appendix K). The original bands were fixed before any cell ran, and the later criterion change is recorded in Appendix I. The construction starts from the untrained frozen-tower initialisation, so the parent is both denoiser and tower, and it adds two options to the sampler. The first option is the acceptance order. It commits the leftmost masked position each round, in place of the most confident one. The second option is masked-key suppression. It hides from each query the keys of masked positions, of committed positions after it, and of the anchor slot. These are the two options the checks below turn off and on. One more setting is fixed by the model and is never toggled. Each model reads its next token where it was trained to read it, so the parent reads at the last committed position and the trained final at the position itself. The witness also stops at the end-of-text token.

Two checks were registered. First, with both options off, the trained frozen final reproduces its committed greedy 42.07 exactly. All 164 generations are byte-identical to the committed run, on a different box, driver and torch build. Second, with both options on, the untrained witness scores 53.05 greedy pass@1 on HumanEval. That equals the parent's committed own-decode score, 53.05, to the digit.

The parent has three greedy scores in this appendix, and they differ. Served by vLLM it scores 53.66, which is what the band was written against. On the frozen harness's own decode path it scores 53.05,

measured and committed before the band was set. Re-decoded on the witness's own box with the same script it scores 52.44. The witness matches the own-decode score to the digit, falls outside the registered band of 53.66 ± 0.5, and sits 0.61 points above the same-box score. After the first measurement the reference was corrected to the own-decode score, and that correction is reported as a deviation (Appendix I).

Token identity was replaced by an agreement criterion before any later cell ran. The witness's committed tokens agree with the parent's teacher-forced argmax on 99.61% of 22,632 positions. The parent's own free-run decode agrees with its argmax on 99.50%, so two runs of this parent do not agree token for token in bf16, and the witness clears the parent's own level. The proposition establishes exact equivalence under its assumptions. The implementation checks show close numerical agreement.

Under sampling (n=20, temperature 0.8, top-p 0.95) the witness reaches 89.56 pass@10 against the parent's 89.23. The registered check was that this score fall inside the parent's paired-bootstrap interval around 89.23. That interval could not be formed, because the parent's cell is a committed summary with no per-problem file. The check reported instead is that the parent's value lies inside the witness's own 95 % problem-resample interval $[85.21, 93.50]$ (Appendix I).

The sweep. Committing K positions per round with zero training gives Table 6. The schedule is exact at every K: $32/K$ rounds per block and K tokens per forward. Only K=1 scores. Greedy pass@1 is zero at every tested K above 1; sampled pass@10 is also zero at K=4, the only other K sampled. The second position in a round is read from a slot whose predecessor has not been committed yet, and the untrained parent has nothing to put there. Commit entropy rises with K, from 0.33 nats at K=1 to 2.42 at K=32.

Table 6: **The untrained witness against tokens committed per forward.** Greedy pass@1 on HumanEval, masked keys suppressed, S=32, 164 problems per cell. Sampled pass@10 (n=20) was measured at K=1 (89.56) and K=4 (0.00). No band was placed on this curve.

K	greedy pass@1	rounds per block	tokens per forward	commit entropy (nats)
1	53.05	32	1	0.33
2	0.00	16	2	1.53
4	0.00	8	4	1.65
8	0.00	4	8	1.66
16	0.00	2	16	1.92
32	0.00	1	32	2.42

Applying the witness decoding rules to the trained model. Before the run we set down one quantity to report at K=4, the gap between the untrained witness and the trained model's 71.60. The two numbers do not separate what the architecture gives from what the training gives. One model is trained and the other is not, and they decode under different rules. The trained model's own sampler commits about 3.8 tokens per forward on average, while the witness rule commits exactly four. Under the witness rules the trained final scores 0.00 at K=4, greedy and sampled. Toggling the two options one at a time on the trained final (greedy, K=4) separates them. Masked-key suppression alone takes its 42.07 to 0.61. The left-to-right order alone takes it to 7.32. Both together take it to zero. The toggles show sensitivity to the tested decoding changes and do not establish why one costs more than the other. The trained model was trained with the keys that suppression hides visible to every query. The tested witness decoding settings do not preserve the trained checkpoint's generation performance, so the existence result of Proposition 1 does not establish compatibility with the sampler used after denoising training.

C THE DENSE-PARENT MATCHED COMPARISON

The dense-parent arms are read in §3, which states every number below in prose; the table is reproduced here for reference.

Table 7: **Dense-parent matched comparison.** Each arm uses 500 M supervised tokens with the same corpus, seed, and evaluation harness; each is evaluated with its native decoder. The "%" column gives the frozen-tower score as a percentage of the parent score. The prespecified replication criterion is met: $r_s = 0.615 \geq 0.45$ and $r_i = 0.076 \leq 0.15$.

	Qwen3-4B-Base (parent)	FROZEN-TOWER-DENSE	%	INPLACE-DENSE (in-place)
HumanEval pass@10	86.42	53.13	61.5	6.55
HumanEval greedy pass@1	59.76	37.80	63.3	0.61
GSM8K	81.88	69.90	85.4	5.38
MMLU-Pro	46.60	45.60	97.9	32.00
MBPP	67.60	41.00	60.7	0.00
BoolQ	86.80	86.60	99.8	80.20
SQuAD-short EM	81.20	76.00	93.6	42.60
JSON validity	98.20	96.60	98.4	9.80

D OUTPUT LENGTH AND RETENTION

The main text orders tasks by output length and reads degradation along that order. This appendix measures the axis with a proxy which is the parent's generated length: the mean number of tokens the *parent* generates for each task under the frozen harness, counted with the parent's tokenizer. For the two multiple-choice tasks the answer is one token and the harness cap is not binding. GSM8K and MBPP hit the 256-token generation cap in every parent sample, so their length is censored at 256: the observed generated length is censored at the cap. We did not rerun these tasks with a higher cap, so the axis is bounded at 256 for them and their true stopping length is unknown. The association between length and retention is read across tasks. Therefore it is not a causal account of any one task. Retention is defined by the arm's score divided by the parent's on the same task.

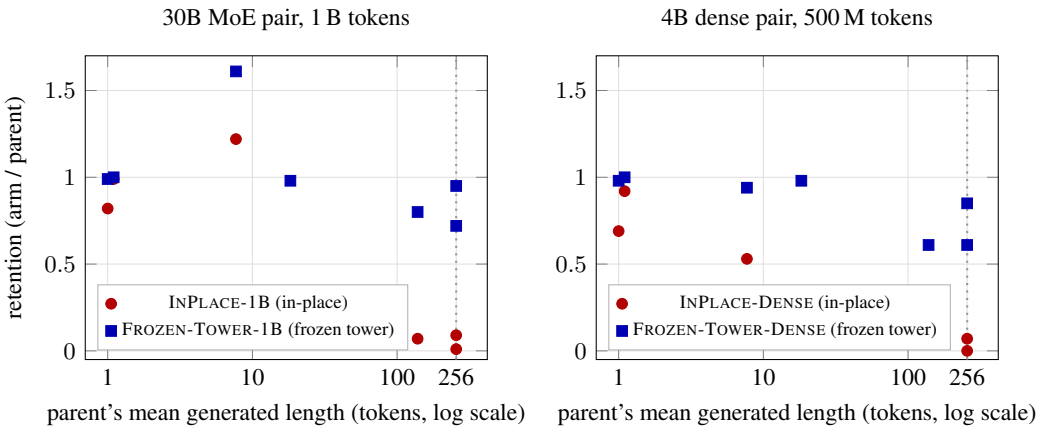


Figure 4: **Retention against the parent's generated length.** Each point is one task (Table 8); the dotted line is the 256-token generation cap, at which generated length is censored. The trend is an association across tasks. Both in-place arms fall with generated length, from 0.69–0.99 at one token to at most 0.1 at 139 tokens and beyond; both frozen-tower arms stay near the parent until the longest tasks. SQuAD-short sits above 1 for the formatting reason given in §3.

Table 8: **The points of Figure 4.** Length is the parent's mean generated tokens; 256^{+} marks the cap. Retention is arm over parent on the score the task is read on (JSON validity for CoNLL-JSON).

		30B MoE pair		4B dense pair	
task	length	in-place	frozen	in-place	frozen
MMLU-Pro	1	0.82	0.99	0.69	0.98
BoolQ	1.1	0.99	1.00	0.92	1.00
SQuAD-short EM	7.7	1.22	1.61	0.53	0.94
CoNLL-JSON validity	18.3	0.32	0.98	0.10	0.98
HumanEval pass@10	138.6	0.07	0.80	0.08	0.61
GSM8K	256^{+}	0.09	0.95	0.07	0.85
MBPP	256^{+}	0.01	0.72	0.00	0.61

Within one task. To read the association without crossing tasks, we split the 164 HumanEval problems into thirds by the length of the canonical solution and score each arm per third, using the mean of its three re-decoded seeds from Appendix E (both arms on one Blackwell host). Table 9 gives the result. The in-place arm solves problems almost only in the shortest third. The frozen-tower arm falls with length and stays far above it in every third. This check was added after review and was not prespecified. The parent has no per-problem file, so retention per third is not formed. Canonical-solution length may also correlate with problem difficulty, so this analysis does not isolate a causal effect of output length. Solution length is the number of lines of the canonical solution after trimming, blank and comment lines included; problems are ordered by that count with ties broken by task id and split into thirds, so a problem with three or six lines can fall on either side of a boundary. No intervals were computed. The script and the group membership will be released with the code.

Table 9: **HumanEval pass@10 by canonical-solution length**, thirds of 55, 55 and 54 problems with median lengths of 1, 5 and 8 lines; each arm's per-problem estimate is the mean of its three decode seeds.

third by solution length	frozen tower (FROZEN-TOWER-1B)	in-place (INPLACE-1B)
shortest third	81.65	13.43
middle third	70.28	0.00
longest third	50.18	0.31

The figure leaves two things open. CoNLL-JSON is the structured task and also a mid-length one, and it sits lowest for its length in both families. These tasks therefore cannot separate a length effect from structured output failing before free text does. The uniform-state arm of Appendix H speaks to the structured-output reading.

The in-place arms are not failing because they run out of room. On GSM8K and MBPP they generate 190 and 152 tokens on average, while the frozen arms use the full 256. Part of the failure at length is stopping early.

Fixed-length canvases are a known weakness of block-wise diffusion. Block diffusion (Arriola et al., 2025) allows variable-length generation, DreamOn (Wu et al., 2026) adds length-control states so a canvas can grow or shrink, and FLUID (Ma et al., 2026) adapts the denoising stride to local information density. Our length result is about retention of the parent's ability at a fixed budget under a fixed canvas, and none of these mechanisms was tested here.

E SAMPLING DISPERSION: PER-CELL TABLES

Every score in this paper other than the greedy rows is an estimate from a finite sample, and the committed values are single draws. We measured the dispersion directly: three decode seeds (1234/2345/3456) on each of five committed cells, with the training and the checkpoints untouched. Greedy generations were token-identical across the three tested decoding seeds on all 164 problems in each checked cell (score range 0.00), so the sampler is the only source of movement in these cells.

Forming the pair ratio within seed and within box gives 12.2–18.7×. The width comes almost entirely from the denominator: the in-place arm's three-seed range is 1.86 on a mean of 4.61, while the frozen-tower arm contributes 1.76 on a mean of 67.47. A ratio taken against a near-floor baseline is not a stable point estimate, and precision on the numerator does not repair it. The committed scores sit below every per-seed ratio we measured, so the committed ratio is below the three re-decoded ratios. Dispersion is a property of the arm rather than of the budget: at 500 M the two training seeds give ranges of 0.58 and 2.24 on the same task under the same protocol, and it cannot be assumed from one checkpoint.

Bootstrap intervals over problems. Decode dispersion measures one source of uncertainty, the sampler, on a fixed set of problems. The set itself is a second source, because HumanEval has only 164 problems in it. Resampling problems with replacement (10,000 draws) and recomputing pass@10 from each cell's per-problem counts gives the 95 % intervals in Table 10. Their half-width is about six points for every cell above 20, twice the three-point single-draw rule. The two quantities answer different questions. The decode resampling describes how far a score moves when the sampler is redrawn on the same problems. The bootstrap interval bounds how far a score on these 164 problems can sit from the score on the population they sample. The comparisons this paper draws are paired on the same problems, so the interval that matters for each is the paired bootstrap of the difference in the lower block. The pair gap and the dense gap exclude zero by a wide margin. The $S{=}16$ stage and the second training seed do not, which is the result reported for both in §3. Intervals are computed from the committed per-problem files, and the committed scores are unchanged.

Table 10: **Bootstrap 95 % intervals over the 164 HumanEval problems.** Upper block: single cells with per-problem counts. Lower block: paired differences, resampling problems once for both arms; where a cell has three decode seeds, the per-problem estimate averages them. The in-place 1 B row uses the three re-decoded seeds of Table 13, since the committed draw has no per-problem file.

cell	pass@10	95 % interval
frozen tower, 1 B (FROZEN-TOWER-1B)	71.60	[65.6, 77.6]
in-place, 1 B (INPLACE-1B), three-seed re-decode	4.61	[2.1, 7.7]
dense frozen tower, 500 M (FROZEN-TOWER-DENSE)	53.13	[46.5, 59.9]
dense in-place, 500 M (INPLACE-DENSE)	6.55	[3.3, 10.2]
placement arm, 250 M	38.29	[31.8, 44.7]
1.7 B donor @ 1.136 tok/param	11.12	[6.9, 15.8]
1.7 B donor @ 0.568 tok/param	6.05	[2.7, 9.9]
1.7 B donor @ 500 M	0.61	[0.0, 1.5]
comparison	**difference**	**95 % interval**
frozen vs in-place, 1 B (3 seeds each)	+62.9	[+56.8, +68.8]
dense frozen vs dense in-place, 500 M	+46.6	[+40.1, +53.2]
$S{=}16$ stage vs $S{=}32$ arm (3 seeds each, Hopper reads)	+1.9	[−1.9, +5.8]
training seed 2 vs seed 1, 500 M (3 seeds each)	+1.8	[−2.3, +5.8]
1.7 B donor, 1.136 vs 0.568 tok/param	+5.1	[+2.1, +8.6]

HumanEval+ and pass@1 for the pair. The same committed generations score on HumanEval+ (Liu et al., 2023), whose test suites are stricter, and at pass@1 (Table 11). The pair scores 63.29 and 6.19 pass@10 on HumanEval+; at pass@1 the ratio is about 20× against 11.6× at pass@10, so the advantage is present per sample and larger there.

Table 11: **HumanEval+ and pass@1 for the pair**, from the committed generations. The in-place committed cell has no per-problem file, so its interval is not computed; its three decode seeds give pass@1 of 1.22–1.52 with intervals inside [0.3, 2.7].

arm	pass@10	HumanEval+ pass@10	pass@1 [95 %]	HumanEval+ pass@1
FROZEN-TOWER-1B (frozen tower)	71.60	63.29	33.54 [28.7, 38.5]	28.38
INPLACE-1B (in-place)	6.19	6.19	1.65	1.49

GSM8K and MBPP intervals. The two large tasks were single draws. Re-scoring the committed generations through the frozen harness and resampling problems (10,000 draws) gives Table 12: the half-width is about two points on GSM8K (1,319 problems) and four on MBPP (500), and the paired frozen-versus-parent gap on GSM8K excludes zero. Every cell first had to reproduce its committed accuracy to the digit; two MBPP cells re-scored one problem differently on this host, in opposite directions, and are shown at their committed values without an interval. The in-place 1 B arm's generations were not archived, so its committed single-draw values stand without an interval. The committed scores are unchanged.

Table 12: **Bootstrap 95 % intervals over problems for GSM8K and MBPP**, in percentage points, from the committed generations. †Re-scored one problem in 500 differently from the committed value; no interval. ‡The parent's committed figures differ between sources: the parity files carry 88.02 / 74.40, the results documents the main text quotes carry 87.19 / 74.60, and the re-scored generations give 88.02 / 74.60; every difference lies inside the interval shown. §No problem was solved in these cells, so resampling problems gives a degenerate interval. The exact binomial 95 % interval for 0 of 500 is [0.0, 0.7].

cell	GSM8K [95 %]	MBPP [95 %]
frozen tower, 1 B (FROZEN-TOWER-1B)	82.56 [80.5, 84.5]	53.40 [49.0, 57.8]
AR parent (Qwen3-30B-A3B-Base)‡	88.02 [86.3, 89.8]	74.40†
dense frozen tower, 500 M (FROZEN-TOWER-DENSE)	69.90 [67.5, 72.3]	41.00 [36.8, 45.2]
dense in-place, 500 M (INPLACE-DENSE)	5.38 [4.2, 6.7]	0.00 [0.0, 0.0]§
placement arm, 250 M	64.29 [61.7, 66.9]	14.00 [11.0, 17.2]
1.7 B donor @ 1.136 tok/param	0.83 [0.4, 1.4]	0.60 [0.0, 1.4]
1.7 B donor @ 500 M	2.35 [1.6, 3.2]	0.00 [0.0, 0.0]§
RND1 at T=0.01	21.00 [18.8, 23.3]	19.00†
comparison	**difference**	**95 % interval**
frozen 1 B vs parent, GSM8K	-5.5	$[-7.6, -3.3]$
dense in-place vs dense frozen, GSM8K	-64.5	$[-67.2, -61.8]$
dense in-place vs dense frozen, MBPP	-41.0	$[-45.2, -36.8]$

Two different Hopper hosts score the same checkpoint's greedy at 42.68, the same to the digit, and a Blackwell host reads 1.83 points away on that same cell. So the difference goes with the architecture, and two hosts of the same architecture agree. Every three-seed comparison in this appendix is formed within one accelerator architecture. A shift measured on one checkpoint says nothing about comparing two different checkpoints across hosts.

Three decode seeds (1234/2345/3456) per cell, training and checkpoints untouched, ranges rather than parametric intervals, every seed reported in Table 13, including anomalous ones. The frozen-tower 1 B and in-place 1 B cells were measured on one Blackwell host so that the per-seed ratio is formed within box. The 500 M and S=16 cells were measured on Hopper hosts. Their comparators are Hopper reads too. The S=16 stage is compared with the S=32 arm's three Hopper draws, mean 68.49. The two training seeds at 500 M were read on the two Hopper hosts that agree to the digit on greedy. None of the re-evaluations in this appendix forms a ratio or difference across the two architectures. The committed headline and attach-density comparisons do cross them and are read with that limitation (Appendix K).

Table 13: **Decode dispersion.** "Committed" is the single draw reported in the main text and is not revised. Greedy generations are token-identical across seeds in every cell (164/164 problems).

cell	metric	committed	draws (1234 / 2345 / 3456)	3-seed mean	range
frozen-tower 1 B, S=32 (Blackwell read)	HumanEval pass@10	71.60	68.36 / 66.60 / 67.46	67.47	1.76
frozen-tower 1 B, S=32 (Hopper read)	HumanEval pass@10	71.60	67.51 / 67.90 / 70.07	68.49	2.56
in-place 1 B	HumanEval pass@10	6.19	3.66 / 4.64 / 5.52	4.61	1.86
frozen-tower 500 M, seed 1	HumanEval pass@10	64.04	64.15 / 63.57 / 63.87	63.86	0.58
frozen-tower 500 M, seed 2	HumanEval pass@10	63.63	65.84 / 64.38 / 66.62	65.61	2.24
frozen-tower S=16	HumanEval pass@10	69.66	70.50 / 69.64 / 71.06	70.40	1.42
any cell	HumanEval greedy	—	identical	—	0.00

These are the two comparisons behind the rule in §3, and this is how far they move. The training-seed difference at 500 M goes from $+0.41$ on single draws to -1.75 on three-draw means. The $S{=}32$ against $S{=}16$ difference goes from $+1.94$ to -1.91. Both arms of each comparison were read on the same accelerator architecture, and neither committed number changes.

Deterministic tasks (GSM8K, MMLU-Pro, MBPP at $n{=}1$, temperature 0) never consult the sampler's generator. Re-running one of them under two seeds reproduced the committed output byte for byte, so their dispersion is zero by construction.

The committed baselines are unseeded single draws: the evaluation entry point called no seeding routine, and where a seed did reach a generator its base was the interpreter's per-process string hash. This does not invalidate any committed number, since pass@10 from $n{=}20$ is a valid estimate whether or not the draw was seeded, but the exact generations cannot be regenerated. The hash seed is pinned in every cell reported here and in all later work.

F INTERFACE ABLATIONS: DETAIL

Denoiser size, and the quantity that orders it. We then hold the tower and the attach fixed and shrink the denoiser, replacing the full parent copy (0.919 B trainable) with dense same-family donors. A 0.6 B donor (0.44 B trainable) at 500 M tokens reaches 5.51 on HumanEval pass@10 and 22.0 on MMLU-Pro, having sat at chance (8.2) at 250 M. A 1.7 B donor (1.409 B trainable) at the same 500 M budget reaches 0.61 and 9.2, and is *flat at chance* across its entire second 250 M (9.4 to 9.2) while the smaller donor nearly tripled over the same interval. It reaches this with three times the trainable parameters and a lower final training loss (6.58). Its commit entropy at 500 M, 0.8515, sits within 0.006 of the smaller donor's *250 M* value (0.8571), where that donor had already fallen to 0.7924 by 500 M.

Ordered by *tokens per trainable parameter*, the 1.7 B donor sits at 0.177 and 0.355, and the 0.6 B donor at 0.568 and 1.136. Along that order MMLU-Pro reads 9.4, 9.2, 8.2 and 22.0. HumanEval reads 0.61, 0.61 and 5.51 at three of those points, because the 1.7 B donor's 250 M checkpoint was never scored on it. Only the last point separates from chance on MMLU-Pro, and it is the only HumanEval score above one point. At its full budget the larger donor has seen fewer tokens per trainable parameter than the smaller one had at half of its.

At a fixed token budget, the smaller donor scores higher in this comparison. That comparison mixes donor size with tokens per trainable parameter. Matching tokens per trainable parameter instead removes the mixing, and it measures nothing about convergence. Neither comparison sets a lower bound on the capacity a denoiser needs.

The 1.7 B donor at matched tokens per trainable parameter. Running the 1.7 B donor to 1.136 tokens per trainable parameter, the 0.6 B donor's 500 M position, holds tokens per trainable parameter fixed across the two donors. The run was set up with two co-primary metrics fixed in advance and a conflict rule: if the two land in bands that point in opposite directions, both are reported and no size conclusion is drawn. They do. At the matched point the larger donor reaches 11.12 on HumanEval pass@10 against 5.51 for the smaller donor, inside its SIZE-HELPS band, and 13.4 on MMLU-Pro against 22.0, inside its KNOWLEDGE-FLAT band. We draw no conclusion about donor size from this pair, and we report what each metric shows.

The knowledge trajectory adds four points. In the extended run, MMLU-Pro was read at four checkpoints, 12.0, 11.6, 10.4 and 13.4 at 0.355, 0.568, 0.852 and 1.136 tokens per trainable parameter, with chance at 10 and one standard error near 1.5. The first of these is a different run from the donor's own 500 M run in Table 14, which reads 9.2 at the same tokens per parameter: the two runs share the recipe and differ in learning-rate path (Appendix I), and both values sit near chance. The larger donor stays within the neighbourhood of chance across a threefold increase in adaptation, while the smaller donor rose from 8.2 to 22.0 over the same interval. Additional training improves the larger donor's HumanEval score, while its MMLU-Pro score remains below the smaller donor's.

Code generation did switch on. The HumanEval margin over the smaller donor is 6.05 against 0.61 at 0.568 and 11.12 against 5.51 at 1.136, an offset of about five and a half points that is present at the first matched point and does not grow. The secondary metrics divide the same way. JSON validity rises from 15% to 81.4%, CoNLL-JSON micro-F1 from 0.0 to 7.6, and SQuAD-short EM from 2.2

to 7.8; GSM8K (0.83) and MBPP (0.6) are at floor for both donors and resolve nothing. At matched adaptation the larger donor scores higher on the formatting metrics and lower on MMLU-Pro. The comparison does not isolate donor size, because the learning-rate trajectories and the hosts differ.

Three caveats attach. The arm's learning-rate path was not schedule-matched to the smaller donor's. An implementation error ran its first 12,140 steps as a complete cosine to the floor and then re-warmed, so the endpoint learning rate is the designed one and the path is not (Appendix I). The direction of the deviation's effect was not measured. It is a further limitation of this arm's comparison. And this is one donor size at one matched point, so it does not locate a capacity floor. The arm also ran on a different accelerator architecture from the smaller donor, with the micro-batch halved and gradient accumulation doubled (Appendix I).

Table 14: **Denoiser size at a fixed tower and attach**, ordered by tokens per trainable parameter (tok/param). The first four rows are the two donors' own runs; the 800 M and 1.601 B rows are the extended run of the 1.7 B donor, whose own checkpoint at 0.355 tok/param (step 5,059) reads 12.0 on MMLU-Pro against 9.2 for the separate 500 M run, the two runs differing in learning-rate path (Appendix I).

denoiser	HumanEval pass@10	MMLU-Pro	trainable params	tokens	tok/param
0.6 B donor @ 250 M	0.61	8.2	0.44 B	250 M	0.568
0.6 B donor @ 500 M	5.51	22.0	0.44 B	500 M	1.136
1.7 B donor @ 250 M	—	9.4	1.409 B	250 M	0.177
1.7 B donor @ 500 M	0.61	9.2	1.409 B	500 M	0.355
1.7 B donor @ 800 M	6.05	11.6	1.409 B	800 M	0.568
1.7 B donor @ 1.601 B	11.12	13.4	1.409 B	1.601 B	1.136
full parent copy @ 500 M	64.04	54.2	0.919 B	500 M	0.544

Attach sets. Density arms use a uniform stride and are strictly nested: $\{3, 7, \ldots, 47\} \subset \{1, 3, \ldots, 47\} \subset \{\ell : \ell \not\equiv 0 \bmod 4\} \subset \{0, \ldots, 47\}$, and every set contains layer 47. Non-attached layers are canvas-only, with zero-length tower keys and values. Nesting is what makes the sequence a monotone inclusion chain.

Placement at matched density. The density arms above change how many tower layers the denoiser reads, and all of them spread those layers evenly through the tower. They therefore say nothing about which layers are read. A fifth arm keeps the density at half and reads the top 24 layers as one block, $\{24, \ldots, 47\}$. Budget, corpus, seed, recipe and harness are identical to the strided arm, and both ran on the same accelerator architecture.

On the metric the density curve is read on, the two placements differ by 1.8 points: the contiguous arm scores 38.29 on HumanEval pass@10 against 40.09 for the strided arm, and this single-seed comparison does not establish equivalence. On two of the five tasks it matters a great deal. MMLU-Pro falls from 57.00 to 42.40, which is below the quarter-density arm's 43.00, and MBPP falls from 35.40 to 14.00, closer to the quarter-density arm's 6.60 than to its own density. HumanEval+ and GSM8K move by less (35.97 to 31.55, and 66.72 to 64.29).

The two placements differ by 1.8 HumanEval pass@10 points, and this single-seed comparison does not establish equivalence. On MMLU-Pro and MBPP the contiguous placement scores 42.40 and 14.00 against 57.00 and 35.40, so density alone does not fix knowledge decoding, and a uniform stride is doing work that a contiguous half does not. And the metric-specificity reported for the density curve holds for placement too: the density result is a statement about a uniformly strided half measured on HumanEval pass@10, with the uncertainty of the contrasts unmeasured, and not a general one.

The step-0 routing divergence anticipates this where the gate metric does not. The contiguous arm reads 0.7165 against the strided arm's 0.3205, higher than every density point including zero attachment. That ordering matches what MMLU-Pro and MBPP show and contradicts what HumanEval pass@10 shows, which places a second limit on the surrogate: it is not capability-ordered at low density, and it is not gate-metric-ordered across placement.

Rounds per block fall monotonically with density. Rounds per block fall 13.7, 11.54, 10.13, 8.4 across a quarter, half, three quarters and full density on HumanEval. The first-round commit fraction reads 3.1–7.3%, 12.2%, 14.7% and 9–22% at the same four densities. The first and last are ranges across tasks and they overlap the middle values, so this metric does not order the arms. Unlike the headline metric, rounds per block do not plateau between half and three-quarter density: more attached layers give earlier commits and fewer rounds.

A cheap surrogate that only partly works. Routing divergence against the frozen tower at step 0, before any training, is monotone in density from a quarter upward: 0.476, 0.321, 0.227, 0.130. It is not capability-ordered at the bottom, since zero attachment reads below a quarter while scoring far worse when trained, and its shape disagrees with the trained curve, which steps at half density where the surrogate is close to linear. It is usable as a coarse ranking signal at or above a quarter density and is not a substitute for a trained run. One recorded discrepancy stands unresolved: the quarter-density level did not reproduce an earlier measurement of the same quantity on the same geometry.

Training loss does not see this damage. Final training loss varies by about 0.3 across arms whose HumanEval pass@10 spans 19.60 to 47.11. The same observation holds across every arm in this paper and is the reason the gates are defined on held-out task metrics.

Denoiser-size arms, recorded non-identities. The 0.6 B and 1.7 B donor arms ran on different hosts and different architectures, a limitation of every comparison between them. The 1.7 B arm used a smaller micro-batch with proportionally more gradient accumulation to fit its host, holding sequences per optimiser step fixed, so the two arms are not bit-identical in their gradient reduction. Generations from the 1.7 B arm are not degenerate: 3,280 completions, mean 560 characters, one empty, fluent and syntactically valid Python that drifts off task rather than failing to produce text.

G Routing under conversion

Expert routing never collapsed in any of our conversions. We logged it in five contexts: full fine-tuning, frozen-with-alignment, block-structured, uniform-state, and the frozen-tower denoiser copy. The router was trainable in all five. The dead-expert fraction was 0.0 at every logged step, and utilisation entropy stayed at or above 0.99, over roughly 2.75 B supervised tokens in total.

Swapping the attention mask moves routing before any training happens. At identical weights at step 0, the divergence is 0.211 under a full-bidirectional layout and 0.121 under a block-causal layout on the small parent, and 0.146 on the 30B parent. All three reproduce across hosts to four decimal places. Block structure therefore keeps routing close to where the AR model had it, and full bidirectionality moves it about twice as far. We are not aware of prior published measurements of routing behaviour under A2D conversion.

H Interventions that did not improve decoding

Three interventions aimed at faster or better decoding were measured, and none of them moved the quantity it targeted.

Few-step distillation does not buy sampler rounds at this scale. Self-distillation through time was run at two grid depths against the deployed adaptive sampler, whose operating point is 4.91 rounds per block. A grid roughly six times above the operating point gave 4.94 rounds, and a rung placed directly *on* the operating point gave 4.99. The speed metric is insensitive to grid depth, and both rungs drift marginally upward. Quality was never the obstacle: the on-operating-point rung scored 71.94 on HumanEval pass@10 against a prespecified floor of 62. We ran the second rung to test an explanation of the first. Uniform ancestral rungs optimise a schedule the adaptive sampler never visits, so a rung placed on the operating point should have moved the round count. It did not, so that explanation does not hold. We report round counts only, because the wall-clock figures from these runs came from a different serving stack than §6 and are not comparable to it.

Self-correction is real and buys nothing. The uniform-state arm is the only one of ours with a live self-conditioning module. Switching it off changes the decode a lot. Revisions per completion

fall from 9.7 to 0.6, and final canvas entropy falls from 0.524 to 0.257. The module is doing exactly what it is for. The arm still scores 0.0 on code generation. A self-correction channel is therefore not sufficient for the commit behaviour that makes a diffusion decoder fast.

More sampler steps buy confidence without quality. Sweeping the step cap on that arm at a fixed entropy floor moves final canvas entropy monotonically, 0.635 at a cap of 16 to 0.363 at 256, at a fivefold cost in throughput, 101.8 to 20.4 tokens per second. HumanEval pass@1 and pass@10 are 0.0 at every cap. The decoder becomes more confident about text that does not solve the task.

I PRESPECIFIED BANDS AND DEVIATIONS

Each arm was launched against a document that fixed its numerical bands, the reading for each band, the launch bars and the stopping rule. The document was committed before the first training step. We never revise a committed number. A later result changes the wording and leaves the number as measured. Four runs departed from what their document specified. We list all four below.

1. *Micro-batch on the 1.7 B donor.* The recipe did not fit the memory of its host, so we halved the micro-batch and doubled gradient accumulation, holding sequences per optimiser step fixed. Schedule, optimiser, freeze set, attachment, data order and seed are untouched. The accumulation order differs from the 0.6 B arm's, so the two arms do not reduce gradients bit-identically.

2. *Learning-rate path on the 1.7 B donor at matched adaptation.* We ran it in two parts so that a report point could be read before the last 15 h were spent. The scheduler received the first part's step count, annealed to the floor over 12,140 steps and re-warmed on resumption. The run ended at the designed learning rate and reached it by a different path. We did not measure which way this moved the arm (Appendix F). We did not rerun the arm, because no conclusion in the paper depends on it.

3. *The witness reference and criterion.* The band for the untrained frozen-tower sampler was written against the parent's served greedy score, 53.66. On the frozen harness's own decode path the parent scores 53.05, measured and committed before the band was set, and re-decoding it on the witness's box gives 52.44. The witness landed on 53.05. It matches the committed score, sits 0.61 above the same-box one, and falls outside the band, so the run stopped there. Before any later cell ran, we replaced the criterion with forced-token agreement against the parent's teacher-forced argmax along the witness's own completions, 99.61% for the witness against 99.50% for the parent's own free-running decode, together with a score within 0.5 of the own-decode reference (Appendix B).

   The criterion also covered sampling. The witness's pass@10 was registered to fall inside the parent's paired-bootstrap interval around 89.23. The parent's cell has no per-problem file, so that interval could not be formed and the registered check could not be evaluated. The gate document, written after the cell was measured, reports instead whether the parent's value lies inside the witness's own problem-resample interval. We did not register that substitute in advance.

4. *An added baseline row.* The parent initialisation scored under each arm's objective (Table 5) was added after the held-out readings, so that each arm's reduction from its own initialisation could be reported. It carries no band and changes no reading.

Bands referenced from the text. The trainable-tower control (§4) was read on two metrics. HumanEval pass@10 at 500 M was read against 64.04, *helps* at or above 67.0, *matched* between 61.0 and 67.0, and *hurts* below 61.0. MMLU-Pro was read against 54.2, *up* at or above 57.2, *flat* between 51.2 and 57.2, and *down* below 51.2. If the two point in opposite directions, both are reported and no single conclusion about freezing is drawn. The witness (Appendix B.3) was read on greedy pass@1 within 0.5 of 53.66, and on sampled pass@10 inside the parent's interval around 89.23, which could not be evaluated (item 3 above). No band was placed on the K-sweep, and the 250 M trajectory points carry no band.

J THE FROZEN HARNESS

Every model in this paper, including the parents and the external baseline, was scored by one harness, fixed before the run and unchanged throughout. Where a comparison crosses implementations we say so explicitly and report both sides (§5).

Task suites. We score two suites. The first holds HumanEval (Chen et al., 2021) (164 problems), HumanEval+ (Liu et al., 2023), MBPP (Austin et al., 2021b) (500), GSM8K (Cobbe et al., 2021) (1,319) and MMLU-Pro (Wang et al., 2024) (500 questions, where chance is 10). The second holds BoolQ (Clark et al., 2019), SQuAD-short exact match (Rajpurkar et al., 2016) and CoNLL-to-JSON validity (Tjong Kim Sang & De Meulder, 2003). We added the second suite because its tasks want short answers, and comparing them with the first suite's longer generations is what makes the output-length effect of §2 visible.

The task definitions are fixed in the harness configuration. HumanEval draws 20 samples per problem at temperature 0.8 and top-p 0.95 with 256 new tokens, plus one greedy sample, and HumanEval+ scores the same generations. GSM8K is the full test split, greedy, 256 new tokens. MBPP is the 500 test problems, 3-shot in the format of Austin et al. (2021b), greedy, 256 new tokens. MMLU-Pro is a fixed 500-question subset, drawn as a seeded uniform sample (seed 42) over the test split in question-id order, with five shots from the validation split in the question's category, direct answer, greedy, 32 new tokens. BoolQ is 500 validation questions drawn with seed 42, 3-shot, 4 new tokens, scored on the first yes/no word, with unparseable answers counted wrong. SQuAD-short keeps the SQuAD v1.1 validation questions whose reference answer has at most five words, draws 500 with seed 42, and scores exact match under SQuAD normalisation after 3-shot prompting with 16 new tokens. CoNLL-to-JSON draws 500 sentences from the CoNLL-2003 test split with seed 42; after 3-shot prompting the model emits one JSON object with PER, ORG and LOC lists in 48 new tokens, validity is whether the first JSON object parses, and micro-F1 is over (type, surface) pairs. Dataset revisions are pinned in the configuration. The questions scored, the generation limits and the scoring rules are also recorded with the committed per-problem files.

Sampling. pass@10 is estimated from n=20 samples at temperature 0.8. Greedy pass@1 is a separate deterministic run and is reported separately throughout, never mixed into a sampled figure. GSM8K, MMLU-Pro and MBPP are scored at n=1 and temperature 0, so the sampler's generator is never consulted and their dispersion is zero by construction (Appendix E).

Decode configurations. The frozen-tower models decode under the confidence rule at $\gamma = 0.8$ with block size S as trained, with the tower resident. In-place models decode under their native block backend. The uniform-state arm decodes under its own entropy-bound sampler; the one cross-application of that sampler to block-trained weights is marked in Table 1 and is what isolates the decoder from the corruption process. The external baseline is scored three ways: as published, under our decoder, and under its own released inference code at a pinned revision, recorded with the code release.

Parent-agreement diagnostic. Every cross-protocol table carries the AR parent measured under the same protocols as the diffusion models. The parent is the control: a protocol change that moves a diffusion model's score while leaving the parent's unmoved is evidence of decoder coupling, and does not exclude model-specific formatting or implementation interactions. In this paper the parent moves by less than three points across the three protocols, and by at most 1.64 points between our decode and the baseline's own code, on HumanEval pass@1. The baseline moves by far more. That is the observation §5 is built on.

Determinism. The committed scores are single draws from unseeded runs: the evaluation entry point called no seeding routine, so the exact generations cannot be regenerated, while pass@10 from n=20 remains a valid estimate. The hash seed is pinned in every cell measured after that finding, and two byte-identical seeded draws were required before any dispersion cell ran. Greedy scores reproduce to the digit across seeds, across processes and, for one checkpoint, across two physically different hosts of the same architecture.

K REPRODUCIBILITY

One configuration per run, committed, never edited in place. Each arm has one configuration file, committed before its launch and never edited afterwards. A changed recipe becomes a new file, and its header states what changed against the file it came from. At launch the freeze set is checked against parameter counts measured from the model itself, and a mismatch aborts the run.

Corpus. All 30B-parent arms draw their training data from one packed corpus of 3 B supervised tokens, built from FineWeb-Edu (the 10 BT sample, 70 % by target tokens), MegaMath (megamath-web-pro, 15 %) and the StarCoder slice of OLMoE-mix-0924 (15 %, code languages only), at pinned dataset revisions recorded in the corpus manifest. The matched pair shares one data order and reads the same prefix of the corpus. The second training seed uses a different permutation of the same corpus. The parent's pretraining data is not public, so this is a generic mix and we cannot match the parent's own distribution. Documents are tokenized with the parent's tokenizer and packed document-aware at 256-token alignment, so no canvas block spans a document boundary. The manifest carries the dataset revisions, per-shard checksums and the sequence and token counts, 1,942,309 sequences in all. The two arms of the matched pair share a bit-identical manifest, which fixes the data side of that comparison. Every host verifies the manifest before training starts.

Hosts and environments. Runs span Hopper and Blackwell hosts with pinned dependency sets per architecture. Every cross-host comparison in this paper is named as such, and the one same-checkpoint control that spans two hosts of the same architecture is reported in Appendix E. The committed headline scores of §3 come from two architectures. The ratio formed within one host is the three-seed one of Appendix E, measured on a single Blackwell host. The attach-density differences of §7 are formed across a Hopper and a Blackwell host and are reported with that limitation. Every committed cell was scored on the host that trained it, and the three-seed re-evaluations were scored on the hosts Appendix E names.

Artefacts. For the arms marked in Table 15, the archived record holds training logs, raw generations and per-problem scores. For the rest it holds evaluation summaries, calibration records and run headers. The in-place 1 B arm's committed cells have summaries only, and its 250 M–750 M checkpoints were not archived (Appendix B). Where generations exist, pass@k can be recomputed at any k and HumanEval+ scored from the same generations without re-running a model. Checkpoints are mirrored off-host, then downloaded again and checked against their checksums. Nothing is deleted from a host until that check passes.

Inventory. Table 15 lists every trained arm whose numbers appear in this paper: its training seed, training host, corruption kernel and canvas block schedule, which artefacts the archived record holds, and what kind of estimate the reported number is. The OLMoE arms train on the OLMoE pretraining mixture. Every Qwen-parent arm trains on the packed corpus described above, the dense pair included. The Qwen-parent arms train in bf16 with fp32 master weights and optimiser moments, and the OLMoE full fine-tune used 8-bit paged moments. Every committed score is a single decode draw. Where Table 13 adds three decode seeds, the three-seed mean is reported beside it, and greedy scores are exact under the tested conditions.

Table 15: **Experiment inventory.** The seed is the training data-order seed. Kernel: compl. = the independent per-position masking of the in-place arm, blockwise = the exactly-k masking of the two-tower kernel; the block entry is the canvas schedule. Marks: ✓ in the archived record; rcpt = off-host copy with a checksum receipt in the record; n/v = a final checkpoint that exists in the archive but whose off-host copy was not checksum-verified for this paper; – = not in the archived record; seeds = present only for the three decode-seed re-evaluations. Estimate: SD = committed single decode draw; +3s = three decode seeds also reported (Table 13). Every committed cell was scored on its training host.

arm	seed	trained and scored on	kernel, block	log	final	gens	per-prob.	estimate
OLMoE (i) full fine-tune	42	GH200	uniform mask	–	n/v	–	–	SD
OLMoE (ii) frozen + align	42	GH200	uniform mask	–	n/v	–	–	SD
OLMoE (iii) block	42	GH200	compl., 32→64	–	n/v	–	–	SD
OLMoE (iv) uniform-state	42	GH200	uniform-state, 64	–	n/v	–	–	SD
INPLACE-1B	42	B200	compl., 32→64	✓	rcpt	seeds	seeds	SD, +3s
FROZEN-TOWER-1B	42	GH200	blockwise, 32	–	rcpt	✓	✓	SD, +3s
FROZEN-TOWER-1B, training seed 2	2	B200	blockwise, 32	✓	rcpt	✓	✓	SD, +3s
S=16 stage	42	GH200	blockwise, 16	–	rcpt	✓	✓	SD, +3s
INPLACE-DENSE	42	GH200	compl., 32→64	✓	rcpt	✓	✓	SD
FROZEN-TOWER-DENSE	42	GH200	blockwise, 32	✓	rcpt	✓	✓	SD
attach 25 %	42	GH200	blockwise, 32	✓	n/v	–	–	SD
attach 50 %	42	B200	blockwise, 32	✓	n/v	–	✓	SD
attach 75 %	42	B200	blockwise, 32	✓	n/v	–	✓	SD
placement, 50 % late	42	B200	blockwise, 32	✓	rcpt	✓	✓	SD
donor 0.6 B @ 500 M	42	B200	blockwise, 32	✓	rcpt	–	✓	SD
donor 1.7 B @ 500 M	42	GH200	blockwise, 32	✓	n/v	✓	✓	SD
donor 1.7 B @ 1.136 tok/param	42	GH200	blockwise, 32	✓	rcpt	✓	✓	SD
distillation rung 32	42	n/r	blockwise, 16	✓	rcpt	–	✓	SD
distillation rung 8	42	n/r	blockwise, 16	✓	rcpt	–	–	SD
trainable tower (§4)	42	H200	blockwise, 32	✓	rcpt	✓	✓	SD
witness cells (App. B.3)	untrained init, FROZEN-TOWER-1B	GH200	left-to-right, 32	✓	n/a	✓	✓	SD
AR parent (Qwen3-30B-A3B-Base)	–	–	–	–	–	✓	–	SD (parity files)

Table 16: **Delivered budgets.** Supervised tokens and optimiser steps as recorded in each run-completion record, which identifies the evaluated final checkpoint. The frozen-tower 1 B run's record states its budget as delivered exactly at 10,113 steps; it shares a bit-identical data manifest and step count with INPLACE-1B, so the two arms trained on the same tokens. Nominal budgets are 250 M, 500 M and 1 B; n/r = not in the record; n/a = the arm did no training. Table 15 gives what the archived record holds for each arm. All arms were scored with `eval_harness_v1`. The primary estimate for every reported number is the committed single draw, by the paper's fixed reporting rule, and the three-seed mean is its uncertainty companion.

arm	delivered budget (tokens / steps)
INPLACE-1B	1,000,075,732 / 10,113
FROZEN-TOWER-1B	1.000 B / 10,113
FROZEN-TOWER-1B, training seed 2	499,236,606 / 5,050
S=16 stage	n/r
INPLACE-DENSE	499,313,300 / 5,050
FROZEN-TOWER-DENSE	499,313,300 / 5,050
attach 25 % / 50 % / 75 %	249,143,808 / 2,520 each
placement, 50 % late	249,143,808 / 2,520
donor 0.6 B / 1.7 B @ 500 M	500,079,173 / 5,059 each
donor 1.7 B @ 1.136 tok/param	1,600,368,302 / 16,190
distillation rungs 32 / 8	n/r
OLMoE (i)–(iv)	n/r
trainable tower	500,079,173 / 5,059
witness cells	n/a

Cited baseline figures. The 2.1 T training budget of Nemotron-Labs-TwoTower is the figure in that paper's abstract; its training section reports 1.4 T tokens across the two training phases alone. The ratios in §3, §6 and §9 use the abstract's figure.

L ADDITIONAL RELATED WORK

Diffusion objectives and samplers. We build on D3PM's absorbing and uniform kernels (Austin et al., 2021a), the masked-diffusion line of SEDD and MDLM (Lou et al., 2024; Sahoo et al., 2024), GIDD and the diffusion duality for the uniform state (von Rütte et al., 2025; Sahoo et al., 2025; Deschenaux et al., 2026), and BD3-LM's block-mask algebra (Arriola et al., 2025). Few-step distillation (Deschenaux & Gulcehre, 2025) and discrete flow matching (Gat et al., 2024; Shaul et al., 2025) are the standing routes to the throughput gap of §6. Appendix H measures the first and finds no gain at this scale.