Reproduce, Then Attribute: A Controlled Study of the LLM Training Lifecycle on a Bit-Exact Qwen3-0.6B Reproduction

Yash Bishnoi

July 7, 2026

https://github.com/yashb98/BuildFromScratch

Abstract

Published improvements to language-model training rarely arrive with attribution: gains are reported for bundles of changes, at loosely matched compute, often from single runs, so a reader learns that a recipe won without learning why. Small-scale reproductions, where controlled re-testing is affordable, are rarely held to a controlled standard. This paper carries a bit-exact reproduction of Qwen3-0.6B (fp32 max $|\Delta\text{logits}| = 0.0$ against the reference on the recorded probe) through the full training lifecycle – pre-training, data composition, mid-training, supervised fine-tuning, and reinforcement learning with verifiable rewards – on a single GB10 machine, admitting a comparison only when exactly one variable differs, training FLOPs match within 5%, and the effect is a paired three-seed bits-per-byte (BPB) delta with a 95% confidence interval on two corpora. Under these gates, a modernization bundle's single-run, in-loop -17.9% validation-perplexity headline decomposes into three individually significant architecture effects plus an early-training optimizer effect, while its schedule and z-loss axes contribute nothing measurable; and a premium-data anneal improves code BPB by $+0.2716$ (95% CI $[+0.2641, +0.2792]$, 3 seeds, paired) over an iso-token control that absorbs the learning-rate confound. Three nulls are reported at the same standard: SFT response-masking does not separate from its iso-FLOP control; GRPO at 0.6B confirms a pre-registered null against a random-reward gate (single seed, directional by design); and context extension is gated off by a step-0 diagnostic. The same gates caught the project's own errors, including an in-loop 0.68-perplexity "win" that collapsed to $+0.009$ on a fixed held-out set.

1 Introduction

Training reports for open language models have become admirably detailed, yet their evidentiary unit remains the bundle: a recipe of architecture, optimizer, schedule, and data changes is evaluated as a whole, at whatever compute each configuration happened to consume, often from a single run. When the bundle wins, the reader learns that it won – not which component carried the effect, whether the gain clears seed noise, or whether it would survive a compute-matched control [8]. Adopters then inherit every cost of the recipe – an optimizer's throughput overhead, a data pipeline's engineering – without knowing which of those costs buys anything. The gap is widest exactly where closing it would be cheapest: reproductions of small models are common, but they typically stop at matching weights or loss curves, and controlled re-tests of published components – one variable, matched FLOPs, multiple seeds – remain rare.

This paper works in that gap by doing both halves in sequence: reproduce first, attribute second. The substrate is a single-file reproduction of Qwen3-0.6B-Base [23], verified against the released weights at fp32 max $|\Delta\text{logits}| = 0.0$ with argmax agreement on the recorded probe (Table 2),

so that every subsequent effect is measured on an implementation whose fidelity is a checked gate rather than an assumption. From that substrate the study walks the training lifecycle in order – pre-training attribution, data composition, mid-training, supervised fine-tuning (SFT), and reinforcement learning with verifiable rewards (RLVR) – and holds every stage to one standard of evidence: a single-variable diff at iso-FLOP (matched within 5%), three paired seeds per arm, bits-per-byte (BPB) deltas with 95% confidence intervals on two corpora, and versioned evaluation suites (Section 3).

The entire study ran on one NVIDIA GB10 with a unified CPU–GPU memory pool of ≈119 GB. That constraint sets the scale honestly: the faithful base consumed 1.19B FineWeb-Edu tokens [17] – roughly 2 tokens per parameter, far below compute-optimal [8] – and the ablation cells range from 42M to 151M tokens per seed (Table 3). The consequence is stated once here and repeated wherever it matters: every effect in this paper is a fixed-budget attribution at 596M parameters, and none is a scaling claim. What the small scale buys in exchange is the ability to afford real controls – re-running an entire comparison across three seeds costs hours, not weeks – and an autonomous, contract-governed research loop executed the experiments unattended, with human gates on irreversible actions (Section 3).

The outcome is two attributed wins, three honest nulls, and a catalogue of the project's own caught errors. The specific contributions are:

- **Bit-exact reproduction as a verification gate (Section 5).** The Qwen3-0.6B-Base reproduction (596,049,920 parameters) matches the HuggingFace reference implementation at fp32 max $|\Delta\text{logits}| = 0.0$; a second reproduction, SmolLM2-135M (134,515,008 parameters), passes the same gate and calibrates how much logit noise device kernels alone introduce (Table 2). A faithful baseline trained on the 1.19B-token budget lands at 2.14× the original model's perplexity at ≈30,000× less data – reported descriptively only, as a single-run, in-loop measurement (Figure 2).

- **An evidence-gating methodology that decides what a number may be called (Section 3).** Comparisons are admitted only under a single-variable diff at iso-FLOP; effects are paired three-seed BPB deltas with 95% confidence intervals on two corpora, scored by version-stamped suites; a missing required evaluation or a single-seed design caps a verdict at directional; and pre-registered expectations bind in both directions – including against the authors, as when a pre-registered English null was violated and reported as such (Table 6).

- **A de-confounded modernization bundle (Section 6).** The motivating result – an IMU-1-derived bundle [6] beating the faithful baseline by 17.9% in-loop validation perplexity at matched 1.19B-token budgets, a single run, not suite-stamped (Figure 1) – is decomposed by re-testing its axes one variable at a time. Only the architecture axis is significant (+0.1179 BPB on WikiText-2, 95% CI $[+0.1005, +0.1354]$, 3 seeds, paired, text-lm-v2), and it splits into three individually significant components: value residuals +0.0355 $[+0.0116, +0.0594]$, LayerNorm scaling +0.0337 $[+0.0175, +0.0498]$, and head gating +0.0256 $[+0.0125, +0.0386]$ (Table 4, Figure 3). The WSD schedule and z-loss axes contribute nothing measurable. The NorMuon optimizer [12] is isolated separately as a large early-training optimization-speed effect (+0.4743 BPB at 42M tokens per arm, 95% CI $[+0.4435, +0.5052]$), defended by a matched-config AdamW learning-rate sweep and explicitly scoped to its budget, not to scale (Table 4).

- **Single-variable data and mid-training effects (Sections 7 and 8).** At matched tokens, a dclm-edu corpus [11] improves code BPB over FineWeb-Edu by +0.7034 (95% CI

[+0.6639, +0.7430], 3 seeds, paired, text-lm-v2) – the largest single effect in the study – and a sequence-preserving 50/50 mix retains 83.9% of that code win while preserving English (Table 5). A premium-mix anneal then beats an iso-token continued-pretraining control on code by +0.2716 BPB (95% CI [+0.2641, +0.2792], 3 seeds, paired) while the control barely moves off the base, so the learning-rate-decay-plus-extra-tokens confound is absorbed by construction (Table 6).

- **Negative results at the same standard as the wins (Sections 9 and 10).** Response-masked SFT does not separate from an iso-FLOP continued-pretraining control on a fixed held-out set: the masked comparison and a tokenizer-free full-sequence cross-check disagree in sign, so the verdict of record is directional (Table 7). GRPO at 0.6B confirms a pre-registered null [25]: it beats neither its SFT floor nor a mandatory random-reward gate [18], with training reward flat at ≈0.9% throughout (Table 8, Figure 7; math-acc-v1, single seed, directional by design). Context extension was gated off at step 0 by a pre-registered passkey threshold the base failed at its own trained window (Section 8).
- **A failure catalogue (Section 11).** Seven file-backed failures are reported as symptom, detection, root cause, fix, and guard – including an in-loop 0.68-perplexity SFT "win" that collapsed to +0.009 once re-scored on one fixed held-out token set (Figure 6), and a token-level shuffle bug flagged because its result was physically implausible. None of the seven was caught by re-reading prose; all were caught by the same mechanical gates that validate the positive results.

The remainder of the paper is organized as follows. Section 2 situates the study among open small-model reports, matched-compute practice, and the RLVR debate. Section 3 defines the evidence gates and verdict semantics; Section 4 specifies the model, data, hardware, and evaluation protocol. Section 5 establishes the verification gate and the faithful baseline. Sections 6 through 10 then present the lifecycle results in training order – pre-training attribution, data composition, mid-training, SFT, and RLVR – and Section 11 collects what the gates caught. The paper closes with discussion and limitations, a reproducibility statement (Section 14), and broader impacts (Section 15).

2 Related Work

Open small-LM training reports and reproductions. Small language models increasingly ship with detailed training documentation. Qwen3 releases open weights down to 0.6B parameters [23]; SmolLM2 documents a data-centric, multi-stage recipe at 1.7B and releases its curated datasets [2]; OLMo 2 releases weights, data, code, recipes, and intermediate checkpoints [16]; and MiniCPM develops small-model training strategies systematically [10]. Closest to our setting, IMU-1 trains a 430M-parameter model with a bundle of modernizations – the NorMuon optimizer, QK-norm, per-head gating, value residuals, and LayerNorm scaling, among others – and reports approaching the benchmark performance of models trained on far more data, with per-component ablations [6]. On the data side, FineWeb documents ablation-driven web curation [17], and DataComp-LM provides a controlled testbed in which model-based filtering emerges as the key lever [11]; our data-composition experiment places a dclm-edu arm against a FineWeb-Edu control at matched tokens (Table 5). These reports evaluate recipes as bundles at production scale. We work in the opposite direction: reproduce one released model bit-exactly (Table 2), then re-test which published components survive single-variable, iso-FLOP, multi-seed attribution on a different base, data pipeline, and budget (Figure 3).

Optimizers and baseline-tuning discipline. NorMuon adds neuron-wise normalization to Muon's orthogonalized updates and reports efficiency gains over both Adam and Muon at the 1.1B scale [12], with AdamW [14] the default comparator. Because optimizer wins are routinely attributed to undertuned baselines, our NorMuon-vs-AdamW comparison (+0.4743 BPB on wikitext-2, 95% CI [+0.4435, +0.5052], 3 seeds, paired, iso-FLOP at 42M tokens per arm, text-lm-v2; Table 4) carries a matched-configuration AdamW learning-rate sweep whose entire spread is roughly one tenth of the gap in Table 4, and we scope the result as an early-training optimization-speed signal at one architecture and one budget, not a claim about behavior at scale.

Compute-optimal scaling and matched-compute comparison. Chinchilla showed that at fixed compute, parameters and training tokens should scale together, and it compared models at an equalized compute budget [8]; we follow the FLOP-accounting conventions of Chowdhery et al. [4]. This study inherits the matched-compute discipline rather than extending it: every comparison varies exactly one factor with training FLOPs matched within 5%, and all budgets sit far below compute-optimal (the base run is ≈2 tokens per parameter; Table 3), so effects are reported as fixed-budget attributions, never scaling laws.

Mid-training, annealing, and long-context evaluation. MiniCPM's warmup-stable-decay schedule frames the decay phase as a vehicle for continued training and domain adaptation [10]; OLMo 2 applies a dedicated late-stage mixture during annealing [16], and SmolLM2 rebalances its mixture across stages [2]. Such reports motivate premium-data anneals but seldom isolate the data effect from the learning-rate decay that accompanies it. Our anneal therefore carries an iso-token control under the identical cooldown: the control barely moves off the base, while the premium mix improves code BPB by +0.2716 (95% CI [+0.2641, +0.2792], 3 seeds, paired, 150.7M tokens per cell, text-lm-v2; Table 6). For long context we adopt the passkey retrieval probe of Mohtashami and Jaggi [15] as a per-rung accuracy ladder with Wilson intervals (ecl-ladder-v1; Figure 5), consistent in spirit with RULER's finding that claimed context lengths overstate effective ones [9].

RLVR skepticism at small scale. Recent work questions how much reinforcement learning with verifiable rewards adds beyond the base model. Yue et al. [25] find via large-k pass@k that base models match or exceed their RLVR-trained counterparts, concluding that RLVR sharpens what the base model already contains rather than expanding it; Shao et al. [18] show that randomly assigned rewards produce large MATH-500 gains on Qwen2.5-Math models specifically and urge validation across model families, which motivates our mandatory random-reward control on a Qwen-lineage base; Liu et al. [13] identify optimization biases in GRPO and propose the unbiased Dr. GRPO variant we adopt; and DAPO documents the system-level techniques needed to make large-scale RLVR reproducible [24]. VibeThinker-3B argues the optimistic case, reporting benchmark results at 3B parameters that it claims rival far larger systems [22]. We contribute a pre-registered small-scale data point: Dr. GRPO with DAPO-style group filtering on our 0.6B base beats neither its SFT floor nor the random-reward gate on GSM8K [5] or MATH-500 levels 1–3 [7], with training reward flat at ≈0.9% (Table 8, Figure 7; math-acc-v1, n=1 seed, directional by design), scored under the pass@k protocol of Chen et al. [3] with Wilson intervals.

Positioning. This paper introduces no new training technique. Its contribution is single-variable, iso-FLOP, multi-seed attribution of published techniques at reproduction scale on one machine, anchored by a bit-exact reproduction, together with negative results reported at the same standard

as wins: SFT response-masking that does not separate from its iso-FLOP control (Table 7), the pre-registered GRPO null (Table 8), and a context-extension direction gated off by a step-0 diagnostic.

3 Evidence-Gated Methodology

This study is as much about method as about results. Every experiment was executed by an autonomous research loop on a single machine, governed by machine-checkable contracts that decide what may launch, what may be measured, and what a number is allowed to be called. The notation and verdict semantics defined here are used by all results sections.

3.1 The Unit of Claim

The atomic claim of this paper is a *paired across-seed bits-per-byte (BPB) delta with a 95% confidence interval, reported on at least two corpora, under a single-variable diff at iso-FLOP, scored by a versioned harness against a measured noise floor*. Each ingredient is defined below.

Metric. For a model m and a fixed evaluation corpus C of byte length $B(C)$, tokenized by m's own tokenizer into $x_1, \ldots, x_N$,

$$\mathrm{BPB}(m, C) \;=\; \frac{1}{B(C)} \sum_{t=1}^{N} -\log_2 p_m(x_t \mid x_{<t})\,. \tag{1}$$

Because the denominator is a property of the raw text rather than of any tokenization, BPB remains comparable when tokenizers differ and across corpora, unlike per-token perplexity. The harness scores fixed windows of 204,600 tokens per corpus over 869,710 bytes of WikiText-2 and 843,643 bytes of held-out Python code, at pinned dataset revisions, with 13-gram decontamination against training data; per-checkpoint perplexities and noise floors on these corpora appear in Table 3. Effects must be reported on both corpora: $\Delta_c = \mathrm{BPB}_{\mathrm{base}}(c) - \mathrm{BPB}_{\mathrm{treat}}(c)$, positive meaning improvement.

Single variable at iso-FLOP. Two arms compare only if exactly one variable differs *and* their training FLOPs, under the $6N + 12 \cdot L \cdot H \cdot Q \cdot T$ accounting of Chowdhery et al. [4], match within 5%; comparisons at unequal compute conflate the variable with budget [8]. Token-matched is not FLOP-matched when the architecture or optimizer cost changes, so the check is computed, not assumed, and recorded with each run (the largest recorded mismatch among the pretraining comparisons is a FLOPs-per-token ratio of 1.00043; Table 4). Every comparison states its matched budget in tokens and steps (Table 1).

Paired seed test. Both arms train the identical seed set $\{0, 1, 2\}$, varying initialization and data order. The effect is the difference of arm means with a Student-t 95% interval, $\Delta_c \pm t_{0.975,\nu}\sqrt{\mathrm{SEM}_b^2 + \mathrm{SEM}_t^2}$, where ν is the Welch–Satterthwaite degrees of freedom floored to an integer (conservative at n=3, where a normal quantile would be too narrow). A delta is *significant* only if the entire interval lies on the improving side of zero. A corpus-subsample noise floor – the spread of the baseline checkpoint over three disjoint evaluation subsamples – is retained as a secondary check; deltas below it are labeled not significant. The floors themselves motivate the design: on this suite the code-corpus perplexity floor is large (Table 3), so single-checkpoint perplexity gaps on code are close to meaningless, whereas across-seed BPB intervals are not. Results are quoted as $+X$ BPB (95% CI [lo, hi], 3 seeds, paired).

Versioned harness. Cross-run comparable numbers come only from the evaluation harness, which stamps a suite version into every result: (text-lm-v2) for the BPB/perplexity suite, (math-acc-v1) for the pinned math answer-extractor and verifier, (ecl-ladder-v1) for the passkey context

ladder. Any suite change bumps the version, so numbers with different stamps are never differenced. Perplexities printed by the training loop are labeled in-loop validation perplexity, single run, and are advisory only. This distinction is load-bearing: an in-loop 0.68-perplexity SFT "win" collapsed to +0.009 (95% CI $[0.0045, 0.0139]$) once re-scored on one fixed held-out token set (Figure 6).

3.2 Claim Classes and Verdict Caps

Verdicts take the values *win*, *loss*, *inconclusive*, and *directional*. Every run carries a lifecycle stage (data, architecture, mid-training, sft, rlvr, ...), and a per-stage registry names the required (HARD) evaluation battery for that stage. A missing HARD item forces the verdict to directional regardless of the point estimate – completeness gates significance, not the reverse. A single-seed run is never significance-testable and is likewise capped at directional; the RLVR cohort was run at one seed by pre-registered design and carries exactly this cap (Table 8). Near-chance downstream results are reported with Wilson intervals as no-signal, never as wins, with pass@k computed by the unbiased estimator of Chen et al. [3].

3.3 Control Reuse and Pre-Registration

Baseline checkpoints are reused across experiments as symlinks and re-scored in-cohort, so every number in a comparison comes from the same evaluation pass of the same harness version: the architecture sub-drill reuses the de-confound baselines (Table 4), and the data-mix study reuses the dclm-edu arm (Table 5). Expected outcomes are written down before launch and bind in both directions. The RLVR plan, written 2026-07-01 before the 2026-07-02 launch, predicted a null at this scale [25] and mandated a random-reward control for Qwen-lineage models [18]; the null was confirmed (Figure 7). The mid-training anneal pre-registered an expected English null that the data violated – +0.0157 BPB (95% CI $[+0.0140, +0.0174]$, 3 seeds, paired; Table 6) – and the violation is reported as exploratory rather than promoted to a claim.

3.4 Launch Evidence and the Ledger

No budgeted GPU run starts until its evidence is recorded: a literature-sourced token budget, the arm × seed plan, the confound check, a measured throughput probe with a derived ETA, and a passing smoke test that executes the exact training script for one step on a tiny batch and asserts a finite, descending loss, a checkpoint save-then-reload round trip, and exit code zero. The ledger is the append-only truth store: a single programmatic interface with atomic writes records techniques, runs, verdicts, suite stamps, and costs, and is the deduplication authority for what has already been tried. A resumable state machine executes the stages unattended – preflight, scan, selection, briefing, data preparation, launch, evaluation, verdict – and always ends by writing a digest, while irreversible actions (new model builds, compute spend, publication) remain human-gated.

3.5 Safety Envelope

All training ran on one NVIDIA GB10 with a unified CPU+GPU memory pool of ≈119 GB and no separate VRAM; over-allocation invokes the kernel OOM-killer and can take down the machine, so safety limits are part of the method. Three constraints apply to every job: (i) a per-process allocator guard caps GPU memory at a 0.85 fraction of the pool and is imported before the framework in every script; (ii) an independent watchdog process kills any unattended trainer at a 0.80 pool fraction – strictly below the guard, so the watchdog fires first – and only one trainer may run at a time; (iii) cross-entropy over the 151,936-token vocabulary is computed in chunked fp32,

never materializing the full sequence-by-vocabulary logit matrix. Throughput and MFU figures are reported as approximate throughout, because the device peak for this hardware is an estimate.

4 Experimental Setup

Model. All experiments run on our from-scratch reproduction of Qwen3-0.6B-Base [23]: a pre-norm, decoder-only transformer [21] with 28 layers and hidden size 1,024. Attention is grouped-query [1] with 16 query heads and 8 key–value heads at an explicit head dimension of 128, set independently in the configuration (it is not hidden size divided by head count, which would give 64). Each block applies RMSNorm [26] with $\epsilon = 10^{-6}$ before attention and before a SwiGLU feed-forward network [19] of intermediate size 3,072; Qwen3 additionally normalizes queries and keys per head (QK-norm, an RMSNorm over the 128-dimensional head axis) before rotary position embeddings [20] with base $\theta = 10^6$ and positions up to 40,960. All projections are bias-free, and the input embedding is tied to the output head over a 151,936-entry vocabulary. Recomputed from this configuration, the model has exactly 596,049,920 parameters: 28 blocks of 15,730,944 each, a tied embedding matrix of 155,582,464, and a 1,024-parameter final norm (Table 2). A second reproduction, SmolLM2-135M [2] with 134,515,008 parameters, shows that the verification protocol is not specific to one model family (Table 2).

Tokenizer. We use the Qwen3 byte-level BPE tokenizer with vocabulary size 151,936. Every perplexity in this paper is computed with the model's own tokenizer; cross-tokenizer perplexity comparisons are never made. Cross-corpus effects are reported in bits per byte (BPB), which normalizes cross-entropy by the UTF-8 byte count of the scored text and is tokenizer-independent.

Data. The pre-training corpus is FineWeb-Edu [17] (`sample-10BT`), streamed and tokenized on the fly. The baseline consumes 1,189,478,400 training tokens (18,150 steps at 65,536 tokens per step; Table 1), approximately 2 tokens per parameter – deliberately far below the compute-optimal ratio [8], since the study optimizes for controlled comparisons rather than absolute quality. A held-out validation slice of 300,012 tokens is document-disjoint from training via a seeded hash split and screened by 13-gram overlap decontamination (0 of 451 validation documents dropped); the in-loop evaluator scores 50 windows of 4,096 tokens (204,800 tokens) on this slice. The data-composition experiments add a dclm-edu slice [11] of 150,107,480 training tokens, pinned by content hash (`dbad8ad7`) and decontaminated with the same 13-gram screen against both evaluation corpora (0 documents dropped). Supervised fine-tuning uses OpenR1-Math-220k pairs totalling 125,000,592 tokens. The GRPO prompt set (`grpo-math-prompts-v1`) contains 7,471 GSM8K and 3,490 MATH level 1–3 training prompts after dropping 14 prompts that overlapped the evaluation sets at the 13-gram level; 7 prompts overlapping the SFT set were flagged and kept. The math evaluation sets themselves (`math-eval-v1`) are decontaminated against all 35,924 OpenR1 SFT problems: 0 of 1,319 GSM8K test items and 0 of 500 MATH-500 items are flagged.

Hardware. Everything runs on one NVIDIA GB10 (Grace Blackwell) workstation whose CPU and GPU share a single ≈119 GB unified-memory pool with no separate VRAM. Training uses bfloat16 with `torch.compile` at sequence length 4,096; cross-entropy over the 151,936-entry vocabulary is computed in fp32 chunks, since materializing the full fp32 logit matrix at this vocabulary size is infeasible in the shared pool. A 50-step probe measured 7,167.4 tokens/s compiled versus 3,787.5 uncompiled (a 1.89× speedup), and the full baseline run sustained 7,444–7,482 tokens/s at a flat 52.4 GB peak memory, completing in 2,663.1 minutes. FLOP accounting follows the PaLM

convention [4]. Because the GB10's peak throughput is an estimate rather than a published figure, model FLOPs utilization is reported as approximate only and never as an exact number.

Evaluation protocol. Three versioned suites produce every cross-run comparable number, and each such number carries its suite stamp. (i) **text-lm-v2**: perplexity and BPB on the wikitext-2-raw-v1 validation split and codeparrot-clean-valid, both at pinned dataset revisions (`b08601e`, `4db92d2`), scoring 204,600 tokens per corpus per cell. The harness measures a per-corpus noise floor on the baseline checkpoint: 1.304 PPL (3.73% relative) on wikitext-2 but 280.6 PPL (68.2% relative) on the code corpus (Table 3). Code-perplexity orderings are therefore always quoted with this caveat, and comparative effects are instead reported as BPB improvements with paired 3-seed 95% confidence intervals, where seeds vary initialization and data order; deltas below the floor or with an interval crossing zero are labeled not significant. (ii) **math-acc-v1**: a pinned answer extractor and verifier over GSM8K [5] and MATH-500 levels 1–3 [7], with pass@k computed by the unbiased estimator of Chen et al. [3] and Wilson 95% intervals for pass@1, under a fixed sampling band (8 samples per item, temperature 0.8, 256 new tokens, fixed band seed) on 100 GSM8K and 50 MATH-500 items. (iii) **ecl-ladder-v1**: passkey retrieval [15] across context rungs 512–8,192 at depths 0.1–0.9, with 40 probes per rung per cell and Wilson intervals, an effective-context-length measurement in the spirit of RULER [9]. Trainer-internal perplexities are labeled throughout as in-loop validation perplexity, single run, not suite-stamped, and are advisory only. Comparative arms are single-variable and iso-FLOP by construction: two arms are compared only if exactly one variable differs and their training FLOPs match within 5%, with matched token and step budgets stated for every comparison.

Hyperparameters. Table 1 consolidates the per-stage hyperparameters – optimizer, learning-rate schedule, warmup, weight decay, batch shape, token and step budgets, and seeds – with each value flagged as reported in its source paper, inferred from a published band, or chosen by us where no published value exists at this scale. AdamW [14] is the optimizer at every stage except the treatment arm of the optimizer A/B itself.

5 Bit-Exact Reproduction and the Faithful Baseline

Every comparison in this study is anchored to implementations that are first verified to be exact copies of their published references. This section documents the verification gate, the faithful small-budget baseline trained on the verified backbone, and where that baseline sits relative to the original released model. It makes no comparative claims beyond that descriptive placement.

5.1 The verification gate

The gate is a deterministic forward-logits comparison: the official HuggingFace weights are loaded into our from-scratch `model.py`, a single fixed prompt ("The capital of France is", input shape $[1, 5]$) is run in fp32, and the next-token logits are compared element-wise against the reference implementation, with a pass threshold of $\max|\Delta\text{logits}| < 10^{-3}$ plus argmax agreement. For our Qwen3-0.6B reproduction [23] the recorded result is not merely within tolerance but exact: $\max|\Delta\text{logits}| = 0.0$, relative error 0.0, and both implementations predict token id 12095 (" Paris") (Table 2). We state the protocol's scope plainly: it is one prompt, fp32, forward logits only – it establishes operator-level implementation identity under the loaded weights, not behavioral equivalence over an input distribution. The parameter count of 596,049,920 is the forward-calculation total of the build's

architecture plan, asserted (within 1%) by the build's unit test, rather than a field written by the verification artifact itself.

A second reproduction, SmolLM2-135M [2] at 134,515,008 parameters, passes the same gate on CPU fp32 with $\max|\Delta\text{logits}| = 0.0$ across the final logits and all 30 per-layer hidden states, identical greedy generation on 5 of 5 probe prompts, and a wikitext-2 validation perplexity of 15.371 identical to the HuggingFace reference (Table 2). The same weights evaluated on GPU show $\max|\Delta\text{logits}| = 4.72 \times 10^{-5}$ (and 1.95×10^{-3} in the layer-14 hidden state), attributed to SDPA kernel-dispatch reduction-order differences rather than any weight or architecture discrepancy. This is a useful calibration: logit deltas at the 10^{-5} scale on GPU are device noise, so "bit-exact" is a claim we make only where the measured error is exactly zero.

5.2 The faithful baseline at 1.19B tokens

The faithful baseline trains the verified Qwen3-0.6B backbone from random initialization for 18,150 steps at 65,536 tokens per step – 1,189,478,400 FineWeb-Edu tokens [17], roughly two tokens per parameter, four orders of magnitude below both the original 36T-token budget and compute-optimal prescriptions [8]. The recipe is AdamW [14] with cosine decay from a peak learning rate of 2.4×10^{-3}, bf16, sequence length 4096, seed 0; Table 1 flags each hyperparameter as reported or chosen by us. In-loop validation perplexity descends from 185,810.49 at random initialization – close to the 151,936-way uniform-guessing level expected of an untrained model – through 60.10 at 2,000 steps to a final 28.65 (Figure 1). All values on this curve are in-loop validation perplexity from a single run, not suite-stamped.

5.3 Gap to the original

The released `Qwen3-0.6B-Base`, trained on ≈36T tokens, scores 13.400 on the identical 300k-token FineWeb-Edu validation slice with identical evaluation code (204,800 tokens scored, 50 windows × 4096). Our 1.19B-token baseline at 28.65 therefore sits at 2.14× the original's perplexity with ≈30,000× less data (Figure 2). The earlier matched-compute learning-rate sweep at ≈131M tokens per arm provides a second, smaller-budget point: its best arm (peak LR 2.4×10^{-3}, carried forward to the full run) reached 46.310, a 3.5× gap at ≈275,000× less data, with the 1.7×10^{-3} and 3.0×10^{-3} arms at 46.892 and 49.276. The two ratios belong to two different budgets and are never conflated. Both are descriptive only – single runs, in-loop trainer evals, no variance estimate – and locate the small-budget models on the same axis as the reference; they are not parity claims.

5.4 Suite-stamped scores for the baseline checkpoint

Because the in-loop 28.65 lives on the trainer's own FineWeb-Edu validation split, it is never compared against numbers from other corpora. The independent evaluation harness stamps the same checkpoint at wikitext-2 perplexity 37.01 and code_py perplexity 438.67 (text-lm-v2, 204,600 tokens per corpus), each with a subsample noise floor – 1.30 absolute (≈3.7%) on wikitext-2 but 280.56 absolute (≈68.2%) on code_py, so code-perplexity differences for this checkpoint are read as directional only (Table 3). These stamped values, not the in-loop curve, are the baseline reference points for every cross-run comparison in the following sections.

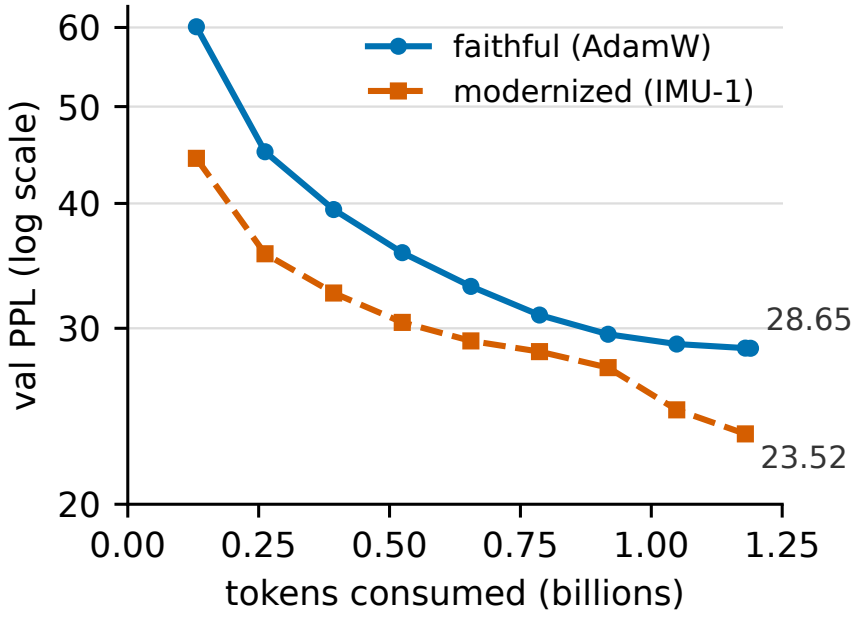

Figure 1: In-loop validation perplexity versus consumed tokens for the faithful baseline and the modernized IMU-1 build [6] at the identical 1,189,478,400-token budget (18,150 steps × 65,536 tok/step, FineWeb-Edu, seed 0). Single run per arm, trainer-internal FineWeb-Edu validation eval, no suite stamp; endpoints are tabulated with their independent suite-stamped scores in Table 3, and no comparative claim is drawn from these n=1 curves here. The faithful curve ends at the post-training eval (28.65); the run completed in 2,663.1 minutes at 52.4 GB peak memory. Sources: `qwen3_baseline2tpp_train.log`, `qwen3_imu1_2tpp_train.log`.

6 Pre-Training: From a Confounded Headline to Attributed Causes

6.1 The motivating result: a bundle wins, but nothing is attributed

With the faithful reproduction verified (Table 2), three builds were trained at matched compute – each consuming 18,150 steps × 65,536 tokens = 1,189,478,400 (≈1.19B) FineWeb-Edu tokens [17] at sequence length 4096 on identical data, schedule shape, and held-out validation set. The *faithful* build follows the published Qwen3 architecture [23]; the *modernized* build applies the IMU-1 bundle [6], a package of recent training changes; the *exploratory* build applies rotary position embeddings [20] to only a 0.25 fraction of each head's dimensions. Table 3 reports the outcome: modernized 23.52 vs faithful 28.65 vs partial-RoPE-0.25 29.54 in-loop validation perplexity (single run per build, not suite-stamped); a fourth arm (partial-RoPE 0.10) died at step 5,450/18,150 and is excluded. The bundle's apparent margin is −17.9% relative perplexity – but the repository itself records that number as n=1 and directional, and this paper treats it as motivation rather than a result: it is a single-seed, in-loop trainer evaluation with no variance estimate.

An independent `text-lm-v2` eval-harness pass on the same three checkpoints (Table 3, right columns) confirms that the *ordering* is not an artifact of the trainer's own eval: on WikiText-2 the modernized checkpoint scores 27.80 vs faithful 37.01 vs exploratory 38.08 (a 24.9% relative gap between modernized and faithful), and the Python-code corpus preserves the same ranking, though its subsample noise floors are 68–92% of the measurement, so the code gaps are directional only. Figure 1 shows the two in-loop validation curves at the identical 1.19B budget.

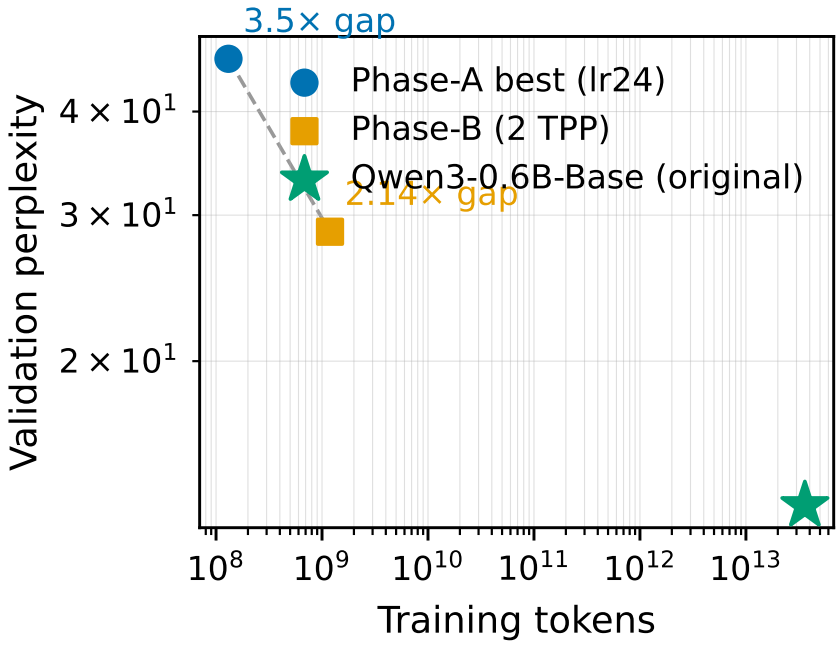

Figure 2: Gap to the original: validation perplexity versus training tokens on the identical 300k-token FineWeb-Edu validation slice (204,800 tokens scored, identical eval code). Original `Qwen3-0.6B-Base` 13.400 at ≈36T tokens; faithful baseline 28.65 at 1,189,478,400 tokens (2.14× gap, ≈30,000× less data); best Phase A learning-rate-sweep arm 46.310 at ≈131M tokens (3.5× gap, ≈275,000× less data). Single run per point, in-loop trainer eval, not suite-stamped; descriptive only. Sources: `original_vs_repro.txt`, `qwen3_baseline2tpp_after.txt`.

The problem is that IMU-1 is a *bundle*: it changes roughly six things at once – the NorMuon optimizer [12], a warmup-stable-decay (WSD) learning-rate schedule [10], a z-loss regularizer [4], and a three-flag architecture package (value-residual connections, layernorm scaling, and attention head gating) [6]. A single n=1 perplexity delta cannot say which of these mattered. The remainder of this section attributes the gain through single-variable, iso-FLOP, multi-seed ablations: first the optimizer in isolation, then a factored de-confound of the remaining axes.

6.2 Optimizer isolation: NorMuon vs AdamW

The optimizer axis was isolated in a single-variable A/B: NorMuon [12] vs AdamW [14] on the 196 2D hidden weight matrices, with embeddings and all 1D parameters remaining under AdamW in *both* arms, so the arms differ only in the update rule applied to the hidden matrices. Each cell trains a fixed 640 steps × 65,536 tokens = 41,943,040 (≈42M) tokens – iso-FLOP by construction – with 3 seeds per arm, paired. Table 4 (top row) gives the result under `text-lm-v2`: NorMuon improves bits-per-byte by +0.4743 on WikiText-2 (95% CI [+0.4435, +0.5052], 3 seeds, paired) and +0.5016 on code (95% CI [+0.4560, +0.5471], 3 seeds, paired), both significant. Two hedges bound this claim. First, it is scoped as an early-training optimization-speed signal at one architecture and one small budget; it is *not* a claim that the advantage persists at scale or to convergence. Second, seeds vary initialization and data-shuffle order on a fixed data split, so the CIs may understate fully randomized run-to-run variance.

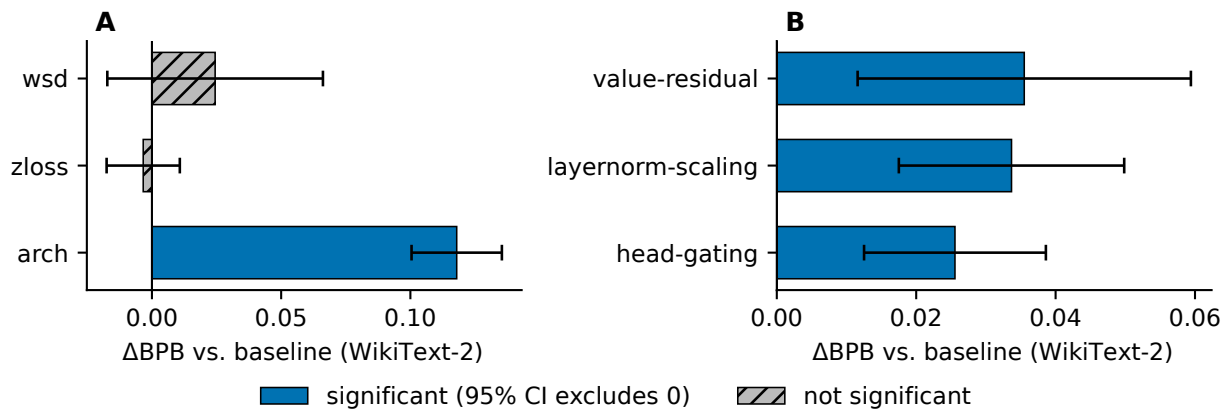

Figure 3: De-confounding the IMU-1 bundle on WikiText-2 BPB improvement over the faithful AdamW baseline (positive = better). Panel A: Phase-1 axes – wsd (schedule), zloss (regularizer), arch (the three-flag architecture bundle). Panel B: Phase-2 sub-drill of the arch bundle into value-residual, layernorm-scaling, and head-gating, scored against the reused Phase-1 baseline checkpoints. Every cell trains 2000 steps × 65,536 tokens = 131,072,000 tokens; n=3 seeds per arm; error bars are 95% CIs; hatched grey bars are not significant. Iso-FLOP holds within the 5% gate for all arms (largest deviation +0.077% parameters, FLOP ratio 1.00043); all numbers are `text-lm-v2`-stamped. Sources: `verdict.json` of experiments `2026-06-18_qwen3-0.6b_imu1-deconfound-p1` and `2026-06-21_qwen3-0.6b_arch-subdrill-p2`, plotted by `fig_deconfound.py`.

A natural objection is an undertuned baseline. A matched-config AdamW learning-rate sweep across 1.7e−3–4.8e−3 (Table 4 caption) spans only ≈0.047 BPB end to end – roughly 10× smaller than the +0.474 gap – so no swept AdamW learning rate comes close to closing it; the result is "NorMuon vs AdamW anywhere in a reasonable LR range," not "NorMuon vs one AdamW point." Caveats: the three off-anchor sweep points are single-seed (only the 2.4e−3 anchor has 3 seeds), and weight decay was held at 0.1 for both arms rather than retuned per optimizer. The speed signal is also not free: NorMuon ran at ≈6,664 tokens/s vs AdamW's ≈7,340 tokens/s in these cells, a ≈9% wall-clock throughput overhead.

6.3 De-confound Phase 1: which axis of the bundle drives the gain?

Phase 1 factors the non-optimizer axes of the bundle – *wsd* (schedule), *zloss* (regularizer), and *arch* (the three architecture flags as one unit) – each ablated single-variable against the faithful AdamW baseline at 2000 steps × 65,536 tokens = 131,072,000 tokens per cell, 3 seeds per arm (12 cells total). The baseline arm was re-verified bit-exact against the canonical model (max $|\Delta\text{logits}| = 0.0$) before launch, and the comparison is iso-FLOP: the arch arm adds only +0.077% parameters, a FLOP ratio of 1.00042, well inside the 5% gate.

Figure 3 (panel A) and Table 4 show the outcome: *only the architecture axis is significant*, with +0.1179 BPB on WikiText-2 (95% CI [+0.1005, +0.1354], 3 seeds, paired) and +0.3049 on code (95% CI [+0.2588, +0.3511], 3 seeds, paired). The schedule and z-loss axes are not significant on either corpus – their CIs straddle zero (Table 4). The recorded verdict is "attributed" with drivers = [arch]: at this budget, neither the WSD schedule nor the z-loss regularizer contributes a measurable quality gain.

6.4 De-confound Phase 2: splitting the architecture bundle

Phase 2 drills into the winning arch axis, ablating its three flags individually – value-residual (vr), layernorm-scaling (ln), and head-gating (hg) – at the same 131,072,000 tokens per cell and 3 seeds per arm (9 new cells). Rather than re-training a baseline, Phase 2 reuses the Phase-1 baseline checkpoints (identical optimizer, schedule, budget, seeds, and data) and re-scores them in-cohort, so every comparison shares one control. Iso-FLOP again holds by a wide margin: value-residual adds 84 parameters, layernorm-scaling adds none, and head-gating adds 458,752 (+0.077%, FLOP ratio 1.00043).

Each flag is individually significant on both corpora (Figure 3, panel B; Table 4): on WikiText-2, value-residual +0.0355 BPB (95% CI $[+0.0116, +0.0594]$), layernorm-scaling +0.0337 (95% CI $[+0.0175, +0.0498]$), and head-gating +0.0256 (95% CI $[+0.0125, +0.0386]$), all at 3 seeds, paired; the code corpus shows the same pattern at larger magnitudes (Table 4). The three single-flag WikiText-2 effects are near-additive: their sum, ≈0.0947 BPB, approaches the bundled arch effect of +0.1179 BPB from Phase 1, leaving only a small residual for interactions and seed noise. The Phase-2 verdict is again "attributed," with drivers = [vr, ln, hg].

6.5 What the headline actually was

The attribution chain replaces one unexplained −17.9% (n=1, in-loop) headline with four small, separately measured causes: an optimizer optimization-speed effect (+0.47 BPB at a 42M-token budget, scoped to early training) and three individually significant architecture effects (value-residual, layernorm-scaling, head-gating; together ≈0.09 of the +0.12 BPB bundled arch gain at 131M tokens) – while the schedule and z-loss axes, plausible-sounding members of the same bundle, contributed nothing measurable at these budgets. None of these attributed effects is a scale claim; all are 3-seed, iso-FLOP, `text-lm-v2`-stamped comparisons at 42M–131M-token proxy budgets on a 596M-parameter model.

7 Data Composition

Data composition is among the highest-leverage levers reported for small-model pretraining [17, 11, 2], and it is also the axis most cheaply isolated: holding the model, optimizer, schedule, and token budget fixed while swapping only the training corpus yields an iso-FLOP comparison by construction. This section tests two treatments against the FineWeb-Edu [17] distribution used for the faithful base: (i) dclm-edu, a curated corpus derived from DCLM [11], and (ii) a 50/50 dclm-edu + FineWeb-Edu mix. Every cell trains the identical 596M model for 2,000 steps × 65,536 tokens = 131,072,000 (≈131M) tokens with 3 seeds per arm; only the data differs, so the train-FLOP ratio across arms is 1.000. Following the control-reuse protocol, the FineWeb-Edu control checkpoints are the reused Phase-1 de-confound baselines (symlinked and re-scored in-cohort), and the mix experiment likewise reuses the dclm-edu arm rather than retraining it. The dclm-edu shards were decontaminated against both evaluation corpora (13-gram overlap; zero documents dropped). All effects below are bits-per-byte (BPB) improvements over the control with paired across-seed 95% CIs (text-lm-v2); Table 5 gives the full cell values.

dclm-edu vs. FineWeb-Edu. On the Python-code corpus, dclm-edu improves BPB by +0.7034 (95% CI $[+0.6639, +0.7430]$, 3 seeds, paired), significant, and by far the largest single effect measured in this study (Table 5). The underlying perplexities make the scale concrete: per-seed code

perplexity falls from ≈1,890 under the FineWeb-Edu control (seed range 1,804–1,962) to ≈250 under dclm-edu (seed range 245–266) — the control corpus simply contains too little code-like text for the model to acquire the distribution. On wikitext-2, dclm-edu is directionally worse, −0.0097 BPB (95% CI $[-0.0255, +0.0061]$, 3 seeds, paired), not significant (text-lm-v2). Pure dclm-edu therefore buys a large code gain at a possible, unresolved English cost.

The 50/50 sequence-preserving mix. The mix arm concatenates equal token counts from the two sources as coherent halves; packing then produces 4,096-token windows each drawn from a single source, and the loader shuffles at the packed-sequence level, so document structure survives mixing. At the same matched 131M-token budget, the mix improves code BPB by +0.5904 (95% CI $[+0.5532, +0.6277]$, 3 seeds, paired), significant — $0.5904/0.7034 = 83.9\%$ of the pure-dclm code improvement, a well-defined ratio because both effects are measured against the same FineWeb-Edu baseline (Table 5). On wikitext-2 the mix shows +0.0161 BPB (95% CI $[-0.0005, +0.0327]$, 3 seeds, paired), not significant but with a positive point estimate: English is preserved rather than traded away (text-lm-v2). The mix is thus best-of-both at this scale — it keeps the English performance the pure treatment put at risk while retaining most of the code win — and it becomes the anneal ingredient for mid-training (Figure 4).

An invalid first run, caught by implausibility. The first mix cohort was invalid: the data loader interleaved the two sources with a token-level shuffle, which destroyed all sequence structure and produced token soup. The evidence gate flagged the failure before any code inspection — the broken mix scored roughly twice the BPB of *both* pure arms on both corpora, which is physically implausible for a blend of two individually healthy distributions. The cohort was discarded, the mixing was rewritten to be sequence-preserving (fix commit `a576baa`), and the arm was retrained from scratch. Section 11 gives the full account, including the diagnostic probe that confirmed the repair.

Bookkeeping. The ledger records both runs with verdict *directional*, a conservative cap; the per-corpus significance keys in each `verdict.json` are the authoritative numbers, and the mix verdict file's overall-verdict string is a recipe-level template that should not be read against them. One protocol deviation is noted rather than papered over: the mix experiment carries no `c5_evidence.json` of its own — its budget and single-variable confound evidence live in `run_arms.sh` together with the reused dclm-experiment `c5_evidence.json`.

8 Mid-Training: Premium-Data Annealing and the Context Gate

Mid-training – the interval between the main pretraining run and post-training – offered two candidate interventions for the 596M base: extending the context window, and annealing onto premium data during a learning-rate cooldown [10, 16]. The first was killed by a cheap diagnostic before any budget was spent; the second is the one stage in this study that reaches a *win* under the per-stage eval-completeness gate. Table 6 summarizes the stage.

8.1 The step-0 diagnostic that gated off context extension

Before budgeting a context-extension run, a step-0 passkey-retrieval diagnostic [15] probed the base checkpoint at and beyond its trained window. The pre-registered gate required accuracy ≥ 0.5 at the trained length of 4096 tokens; the base scored 0.083 (Table 6). Per-length accuracies were 0.333

at 512, 1024, and 2048 tokens, 0.083 at 4096, 0.583 at 6144, and 0.250 at 8192. The diagnostic nominally reports an effective context length of 6144 because the 6144-token rung happens to clear the absolute threshold, but the ladder is non-monotone and the above-trained-window spike at 6144 is better read as instability of a small unstamped probe than as genuine capability; the decision variable is the trained-length accuracy, and it failed by a factor of six. The recorded verdict is that context extension is moot – the base collapses *before* its trained window – so the extension sub-stage stayed propose-only and no extension run was launched. This is the pattern that effective-context benchmarks document at much larger scales, where usable context falls well short of the nominal window [9]; here the gap is severe enough at 596M and 1.19B training tokens that extending the nominal window would optimize the wrong margin.

8.2 Annealing onto the premium mix under an iso-token control

The anneal starts from the faithful base checkpoint (1.19B FineWeb-Edu tokens [17]) and runs a "1-sqrt" cooldown from peak learning rate 2.5×10^{-4} to 2.5×10^{-5} (a 10% floor, 46 warmup steps) for 2,300 steps × 65,536 tokens = 150.7M tokens per cell, ≈13% of the pretraining budget, with n=3 seeds per arm. The treatment anneals onto the 50/50 dclm-edu/FineWeb-Edu mix selected by the data-composition study [11] (Table 5); the control runs the identical cooldown on iso-token FineWeb-Edu. Exactly one variable (the data) differs, so the comparison is iso-FLOP by construction, and the design eats the standard confound of anneal experiments: whatever the learning-rate decay plus 150M additional tokens do on their own is captured by the control.

The answer is that they do approximately nothing. The control moved +0.0050 BPB on code and +0.0036 BPB on wikitext-2 relative to the un-annealed base (text-lm-v2; Table 6), while the mix anneal beat the control on code by +0.2716 BPB (95% CI $[+0.2641, +0.2792]$, 3 seeds, paired), significant (text-lm-v2; Figure 4). The data, not the schedule, carries the effect. Against the constant-learning-rate continued-train sibling on the same mix (+0.5904 BPB on code, 95% CI $[+0.5532, +0.6277]$; an upper-bound reference at a similar 131M-token budget, not an iso-FLOP comparison), the 150.7M-token cooldown retains ≈46% of the effect.

On English, the pre-registration expected an honest null; the measurement violated it: wikitext-2 improved by +0.0157 BPB (95% CI $[+0.0140, +0.0174]$, 3 seeds, paired), significant (text-lm-v2), ≈1.3% relative. This is reported as an exploratory violation of the pre-registered expectation, not as a claim: the point estimate is the same size as the constant-learning-rate sibling's +0.0161 BPB, which was *not* significant, and what pushed the anneal's identical-size effect past significance is the cooldown's much tighter across-seed spread. A same-size estimate flipping significance purely through variance is exactly the situation pre-registration exists to flag.

8.3 The effective-context-length ladder

Both anneal arms and the base were then scored on a passkey ladder over context rungs 512–8192 (ecl-ladder-v1; n=40 probes per rung per cell, arms pooled over 3 seeds, Wilson 95% CIs; Figure 5). The base exhibits an anomalous dip at its own trained length: 4096-rung accuracy is exactly 0.025, far below its 512-rung 0.200. Both anneal arms repair the dip to 0.400 – the FineWeb-Edu control included, so the repair does not require the premium data. The mix arm additionally lifts the pooled 512-token rung to 0.8167 versus 0.20 for the base, and no cell regresses relative to the base at any rung (the recorded regression list is empty). The mechanism of the trained-length dip, and of its repair by an ordinary cooldown, is unexplained; with n=40 probes per rung the ladder characterizes the phenomenon without diagnosing it, and it is flagged as an open question in Section 13. The step-0 diagnostic's independent 4096-token failure (0.083) is consistent with the

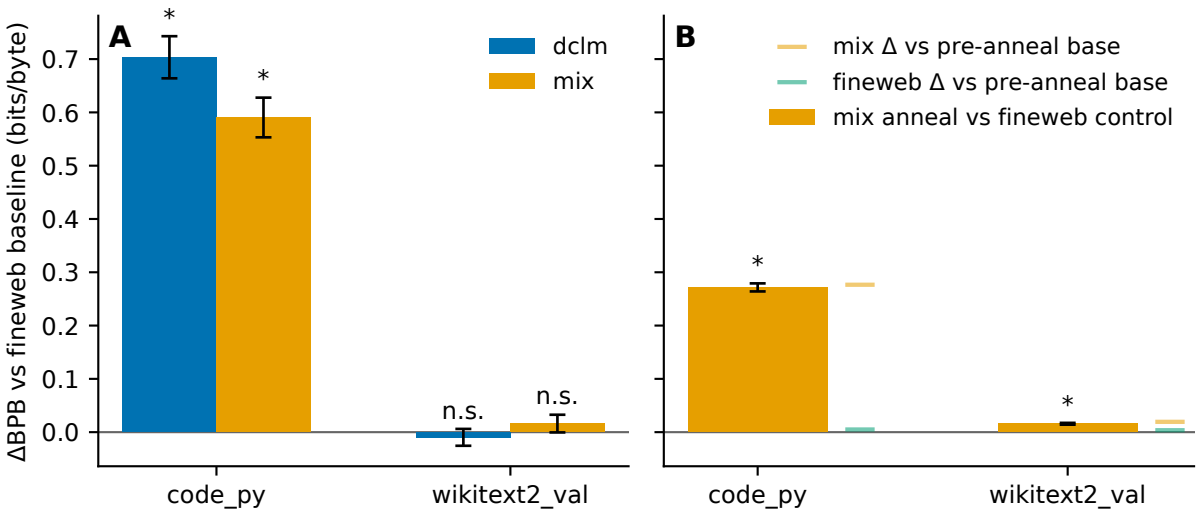

Figure 4: Data-composition and mid-training-anneal effects: BPB improvement over the FineWeb-Edu control (positive is better) with paired across-seed 95% CIs, n=3 seeds per arm, suite `text-lm-v2`, on `code_py` and wikitext-2. Data-composition arms (dclm-edu; 50/50 mix) train 2,000 steps × 65,536 = 131.1M tokens per cell at full LR; the mid-training anneal arms run the 1-sqrt cooldown for 2,300 steps × 65,536 = 150.7M tokens per cell (≈13% of the 1.19B-token pretrain) against an iso-token FineWeb-Edu control. Sources: `verdict.json` of the 2026-06-24 dclm-vs-fineweb, 2026-06-26 data-mix-composition, and 2026-06-30 midtrain-anneal experiments; rendered by `fig_data_anneal.py`.

same anomaly.

8.4 Robustness and the stage verdict

Two integrity checks qualify the headline. First, one treatment cell (mix, seed 0) was twice terminated by an external SIGTERM (exit code 143; not the resource watchdog, whose kill line was never approached) and resumed; the resume rebuilds the data shuffle without fast-forwarding, which if anything disadvantages the treatment. Excluding that cell entirely, the code effect holds: +0.2717 BPB (95% CI $[+0.2333, +0.3100]$), significant, with the caveat that the treatment arm drops to n=2 seeds and the CI widens accordingly; the excluded cell's BPB (1.8521) is not a within-arm outlier (arm range 1.8489–1.8550). Second, the loaded token pools differ by ≈0.7% in the control's favor, and the control lost anyway – the imbalance is conservative with respect to the conclusion.

Under the per-stage eval-completeness gate, mid-training required three hard items – the effective-context-length ladder, short-context non-regression, and the anneal gain against an iso-token control – and all three are present, with the single-variable iso-FLOP check recorded in the run ledger. Mid-training is therefore the one lifecycle stage in this study that is hard-complete under the gate, and its recorded verdict is *win*.

9 Supervised Fine-Tuning: The Win That Was Not

The supervised fine-tuning (SFT) stage poses a single-variable question: does response masking — computing the loss only on response tokens, the standard practice in instruction tuning —

improve a small reasoning-SFT recipe over plain continued pre-training on the same data? The stage produced the study's most instructive de-confound: an apparently significant 0.68-PPL win for masking that turned out to be an artifact of scoring the two arms on different token sets, and that collapsed to roughly a hundredth of a perplexity point under a corrected protocol (Figure 6, Table 7).

Setup. Both arms start from the faithful 1.19B-token base checkpoint and take one pass over the same prepared OpenR1-Math reasoning corpus of 125,000,592 tokens, in the spirit of small-model reasoning distillation [22]: peak learning rate 5×10^{-5} with cosine decay to 8×10^{-8}, 5% warmup, global batch 128 (micro-batch 4, 32 accumulation steps), AdamW [14], n=3 seeds per arm (Table 1). The treatment applies a response mask; the control is an iso-FLOP `-no_mask` continued-pretrain arm that is identical in data, ordering, schedule, steps, and token budget — the loss mask is the only variable. Because prompts are a small fraction of this corpus, the masked arm takes loss on 97.6% of tokens versus the control's 100%; the treatment effect being tested is therefore the exclusion of a thin prompt-token slice from the objective.

The advisory in-loop result. The in-loop evaluation (training-loop perplexity, not suite-stamped, n=3 seeds per arm) showed SFT at 11.595 versus the control at 12.273 — a +0.679 gap with a tight interval (95% CI $[+0.676, +0.681]$) that would read as a decisive win (Table 7, top block). But the comparison is confounded: each arm reused its own training mask at evaluation time, so the SFT arm was scored on response tokens only while the control was scored on all tokens. Perplexities over different token sets are not comparable numbers. The in-loop verdict flagged this itself and capped the result to directional, advisory-only, pending re-scoring under a shared protocol.

The corrected protocol. The verdict of record re-scores all seven checkpoints (base plus three seeds per arm) on one fixed held-out response-masked reasoning set: 66 held-out OpenR1-Math documents comprising 228,529 response tokens, produced by a seeded document-level split with 13-gram deduplication against the training set. On this common footing, base perplexity 14.127 falls to 11.573 (SFT) and 11.582 (control) — improvements of 18.1% and 18.0% respectively — and the 0.68-PPL gap collapses to +0.009 PPL (95% CI $[+0.0045, +0.0139]$, n=3 seeds per arm), nominally significant but a small fraction of the advisory number. A tokenizer-free full-sequence cross-check on the same documents flips the sign: −0.006 PPL (95% CI $[-0.011, -0.001]$) with the control ahead, scored not significant under the pre-specified one-sided improvement test. Since the masked and full-sequence comparisons disagree, the recorded verdict is *directional*: the two arms are not yet separable. Across-seed spread on the held-out set is 0.001 PPL within the SFT arm and 0.004 PPL within the control arm, so the ambiguity is not seed noise — the effect itself sits at the scale of the measurement.

Forgetting probe. Neither arm damages the base distribution. On a held-out FineWeb-Edu probe [17], base perplexity 21.495 rises to 21.652 (SFT) and 21.660 (control), a +0.7% shift (Table 7, bottom block). On the standard suite (text-lm-v2, one representative seed per arm), wikitext-2 moves from 37.010 to 37.082/37.084 against a noise floor of 1.304, and the code corpus from 438.673 to 425.689/425.594 against a floor of 280.564 — both labeled retained/not significant. The three-seed confidence intervals in this section apply to the in-domain reasoning set only; the standard-suite retention numbers are single-seed spot checks.

Interpretation. The real effect at this scale is the *data*, not the masking: continued training on in-domain mathematical reasoning delivers essentially the entire 18% perplexity gain, and excluding the 2.4% prompt-token slice from the loss adds at most a hundredth of a perplexity point of separation, of ambiguous sign. The headline that response-masked SFT beats continued pre-training does not survive an honest iso-FLOP control. The methodological point generalizes: a comparison whose arms are scored on different token sets can manufacture an apparent effect nearly two orders of magnitude larger than the corrected estimate, while presenting a confidence interval tight enough to look unimpeachable. The study's evidence gates treated the in-loop number as advisory by construction; the fixed held-out set, defined once and shared across all arms, is what the verdict rests on.

10 RLVR at 0.6B: A Pre-Registered Null

The final lifecycle stage asks whether reinforcement learning from verifiable rewards (RLVR) adds reasoning capability on top of the SFT checkpoint at this scale. The recent literature motivates skepticism rather than optimism. Yue et al. [25] argue that RLVR sharpens the distribution the base model already supports—raising pass@1 while narrowing rather than expanding pass@k coverage—and Shao et al. [18] show that Qwen-lineage models improve under spurious and even random rewards, so an RLVR gain claimed on this lineage is uninterpretable without a random-reward control. The stage plan (researched 2026-07-01, one day before the 2026-07-02 launch) therefore pre-registered both the prediction and the decision rule: on this FineWeb-Edu→SFT base, which the plan characterizes as roughly 10× undertrained relative to compute-optimal token budgets [8], RLVR is expected to sharpen rather than expand, "the most likely true outcome is a null-or-tiny attributable Dr.GRPO gain that the control battery should correctly refuse to call a win," and a win may be declared only if GRPO beats *both* the SFT floor and a mandatory random-reward gate arm, with "beats" fixed in advance as non-overlapping Wilson 95% confidence intervals.

All arms are decoded on one fixed band: 100 GSM8K test items [5] and 50 MATH-500 level-1–3 items [7], 8 samples per item at temperature 0.8 with at most 256 new tokens, band seed 20260701. pass@1 carries a Wilson 95% CI; pass@8 is the unbiased estimator of Chen et al. [3]; answers are scored by a pinned, version-stamped verifier (math-acc-v1). Contamination was checked before any number was read: a 13-gram index against the 35,924 SFT problems flags zero items in either eval set, and 14 eval-overlapping prompts were dropped from the GRPO training-prompt pool (Table 8).

A cheap Phase-1 gate preceded any RL spend. On this band (math-acc-v1) the SFT checkpoint scores 1.1% pass@1 on GSM8K (Wilson 95% CI $[0.6, 2.1]$) and 7% pass@8, and 1.5%/10% on MATH-500 L1–3; the pretraining base scores 0.6%/5% and 2.25%/14% respectively (Table 8). The pre-registered GO rule was absolute solvability—proceed if and only if the SFT arm solves at least 3 items across the band—and it fired with 12 solved items. What the gate did *not* establish should be stated plainly: GO fired on absolute solvability, not on SFT beating the base. The SFT and base intervals overlap on every cell, and the base beats SFT on MATH-500 pass@8 (14% vs. 10%)—an early warning that arm separations near this accuracy floor sit inside the noise.

Phase 2 trained Dr.GRPO [13] from the SFT checkpoint for 300 steps of 16 prompts × G=8 rollouts (38,400 rollouts total, at a measured ≈362 s per step; roughly 88 hours estimated for the three arms): clip ratio 0.2, k3 KL penalty to the frozen SFT reference with $\beta = 3 \times 10^{-3}$ (a coefficient flagged as inferred, not paper-pinned), learning rate 10^{-6}, and DAPO-style filtering of zero-variance rollout groups [24], which retained 13.49 of 16 groups per step on average (Table 1)—the update was not gradient-starved. The training signal never moved: the per-step verifier-correct rate (the fraction of the 128 rollouts per step scoring exactly 1.0, not the shaped reward) is flat at

≈0.9% (mean 0.0092; first-50-step mean 0.009 vs. last-50 0.008; OLS slope $+5.8 \times 10^{-7}$ per step), and the per-token KL to the reference stays near 1.7×10^{-4} throughout, i.e., the policy barely moved (Figure 7). Two controls ran under the same recipe. The random-reward gate arm [18] replaced the verifier with a content-blind Bernoulli(0.5) draw and trained at reward ≈0.5, confirming that the sampling, advantage, and update pipeline all function. The RFT control spent iso-generation compute: it drew the same 38,400 rollouts, collected the 352 verifier-correct completions (≈0.9% of rollouts, consistent with the GRPO reward floor), and ran one masked SFT pass over 200 of them.

Table 8 reports the final paired evaluation of all four arms on the same band (math-acc-v1). On GSM8K, GRPO scores 1.0% pass@1 / 7% pass@8 against the SFT floor's 1.12%/7%, the random-reward arm's 1.0%/6%, and RFT's 1.5%/11%. On MATH-500 L1–3, GRPO's nominal bump to 2.25%/16% over SFT's 1.5%/10% resembles the expected sharpening story until it is compared with the pretraining base: GRPO's MATH-500 pass@1 of 2.25% exactly equals the base checkpoint's Phase-1 value on the same band, so the "gain" does not clear the model the SFT stage started from. Both pre-registered gates fail on both eval sets: the GRPO intervals overlap the SFT floor's and the random-reward gate's everywhere (`grpo_beats_sft_floor` and `grpo_beats_random_gate` both evaluate to false). The one nominally interesting cell—RFT's 11% pass@8 on GSM8K against the floor's 7%—is not significant (overlapping CIs) and, at one training seed, is at most a hypothesis for future work.

The recorded verdict is *directional*, capped by design rather than by accident: the per-stage evaluation-completeness gate requires at least three paired seeds for any non-directional RLVR call, and this cohort deliberately ran one seed with proceed-to-phase-3 set to false—buying two further seeds would only have tightened a confidence interval around zero. The pre-registered null is confirmed: at 0.6B, on this undertrained base, the measurable reasoning gain arrived with SFT, not with RL, consistent with the sharpen-not-expand account of Yue et al. [25] and the Qwen-lineage spurious-reward findings of Shao et al. [18]. The claim is deliberately bounded. It says nothing about RLVR at larger scales, on stronger bases, or with longer training; it says that at this scale, under the controls the literature itself demands, an honest evaluation battery refuses to call RLVR a win—and that a control battery designed before launch is what makes such a refusal cheap.

11 What the Gates Caught: A Failure Catalogue

Every controlled result in this paper passed through the same mechanical gates: implausibility checks on scored verdicts, a fixed held-out evaluation set required before any win call, iso-FLOP confound checks, and a claim-to-file audit that opens every cited artifact. This section catalogues what those gates caught in the project's own work. Each entry gives the symptom, the detection mechanism, the root cause, the fix, and the guard that remains.

1. Token-level shuffle in the first data-mix run. Symptom: the 50/50 dclm-edu/FineWeb-Edu mix arm scored roughly 2× *worse* than both pure parents (wikitext-2 BPB 2.77 against ≈1.52, as recorded in the fix commit) – physically implausible for a blend, whose loss should sit near the convex hull of its parents. Detection: the implausibility check on the verdict flagged it before any conclusion was recorded. Root cause: a `numpy` shuffle applied at the *token* level interleaved the two sources token by token, destroying all sequence structure. Fix: commit `a576baa` concatenates the two coherent halves and mixes at the packed-sequence level, with the loader shuffling whole 4096-token windows. A 30-step probe after the fix shows the training cross-entropy descending from 11.65 to 8.04, where the token-soup variant had stayed flat near 11. The broken cohort was

discarded and rerun; the corrected mix result appears in Table 5. Guard: the loader now carries an in-code warning, and the implausibility check stays in the scoring path.

2. The SFT eval-token confound. Symptom: in-loop advisory scoring showed response-masked SFT beating its iso-FLOP control by 0.68 perplexity, nominally significant (Figure 6; in-loop values, not suite-stamped). Detection: the protocol forbids a win call from in-loop numbers; a fixed held-out set must be scored first. Root cause: the SFT arm was scored on response tokens only while the control was scored on all tokens – two incomparable token populations. On one fixed held-out response-masked set the gap collapses to +0.009 perplexity (95% CI [+0.0045, +0.0139], 3 seeds, paired), and a full-sequence cross-check gives −0.006 in the control's favor, not significant; the two disagree, so the verdict of record is directional (Section 9, Table 7). Guard: in-loop perplexities are permanently labeled advisory.

3. The RFT smoke-log misread. Symptom: an earlier draft of the RLVR verdict stated that the rejection-fine-tuning control collected zero verifier-correct completions. Detection: the claim-to-file audit for this paper re-read the full training log rather than its opening lines. Root cause: the "0 collected" line belongs to the one-step smoke test of 2026-07-02 at the top of the same log; the real 300-step run collected 352 of 38,400 rollouts (≈0.9%, matching the flat ≈0.9% GRPO training reward; Table 8). Fix: the verdict file and the repository README were corrected on 2026-07-06; the correction note is embedded in the regenerated artifact. The null verdict is unchanged – 352 completions were too few to separate RFT from the floor – but "zero" was wrong. Guard: smoke-test output is now read as a distinct log segment.

4. The README data-ratio pairing error. Symptom: repository documentation paired the 2.14× gap-to-original with "≈275,000× less data." Detection: the claim-to-file audit; the source file pairs 275,000× with the 131M-token Phase-A run, whose gap is 3.5×. Root cause: numbers from two different runs merged into one sentence. The correct pairing is 2.14× at ≈30,000× less data (1.19B tokens) and 3.5× at ≈275,000× less data (131M tokens), as in Figure 2; both gaps are ratios of in-loop validation perplexity, single run each, and are descriptive only. Corrected 2026-07-06.

5. The incomplete partial-RoPE-0.10 arm. The exploratory partial-RoPE-0.10 build died at step 5,450 of 18,150 (≈30% of budget), with a last-eval in-loop validation perplexity of 50.71 (single run, not suite-stamped). It is reported as incomplete in Section 6 rather than silently dropped – an honesty entry rather than a gate catch.

6. A stale derived field in the ledger. Symptom: the ledger entry for the mid-training anneal carries a sub-field `c25_cap` reading "directional" beside a verdict field reading "win." Detection: the audit cross-checked every ledger field against the run's verdict artifact. Root cause: the cap predates the same-night upgrade that added the effective-context-length ladder and made the stage's required evaluation battery complete; the verdict fields were updated, the derived sub-field was not. The authoritative verdict file (no missing hard items) supports the win reported in Table 6. Guard: noted as a standing hazard – derived fields duplicated across artifacts can go stale independently of the record they summarize.

7. A parameter-count citation pointing at the wrong file. Symptom: the model README cited `verify.json` as evidence for the 596,049,920-parameter count (Table 2), but that file contains no parameter field. The number itself is correct – it is recovered exactly by recomputation from

the architecture configuration and asserted in the build's test suite – only the citation was off. Detection: the claim-to-file audit. Guard: every number in this paper traces to the file that actually contains it.

None of these seven was caught by re-reading prose. Each was caught by a mechanical gate: an implausibility check on a scored verdict, a fixed held-out set required before any win call, or a claim-to-file audit that opens every cited artifact. These are the same gates that validate the positive results of Sections 6–10; a protocol that demonstrably catches its own errors is the strongest warrant this study can offer for the claims that survived it.

12 Discussion

What transfers. The portable output of this study is the evidence standard, not the effect sizes. The unit of claim – a paired across-seed bits-per-byte delta with a 95% confidence interval on two corpora, under a single-variable diff at matched train FLOPs, scored by a versioned suite (Section 3) – is independent of scale and cheap relative to the training runs it disciplines. Two empirical regularities also seem worth carrying forward. First, bundle claims should be distrusted by default. The modernized bundle [6] supplied this study's motivating headline, a 17.9% in-loop validation-perplexity gap at matched 1.19B-token compute (single runs, in-loop, not suite-stamped; Table 3); under single-variable iso-FLOP re-tests it decomposed into three small, individually significant architecture effects of +0.0256 to +0.0355 BPB on wikitext-2 (each with a zero-excluding 95% CI, 3 seeds, paired, 131M tokens per cell, text-lm-v2), one large early-training optimizer-speed effect (+0.4743 BPB, 95% CI [+0.4435, +0.5052], 3 seeds, paired, 42M tokens per cell), and two components – the WSD schedule and z-loss – that contributed nothing measurable (Table 4). None of that structure was visible in the bundle run. Second, data interventions dominated at this scale. The largest attributed effect in the study is a data swap: dclm-edu over FineWeb-Edu at +0.7034 code BPB (95% CI [+0.6639, +0.7430], 3 seeds, paired, 131M tokens per cell, text-lm-v2; Table 5), and the mid-training anneal added +0.2716 code BPB over its iso-token control (95% CI [+0.2641, +0.2792], 3 seeds, paired, 150.7M tokens per cell, text-lm-v2; Table 6). The attributed architecture effects are an order of magnitude smaller (Table 4), the optimizer effect is scoped to optimization speed rather than converged quality, and the RL stage returned a pre-registered null: GRPO beat neither the SFT floor nor the random-reward gate, with training reward flat at ≈0.9% (math-acc-v1, n=1 seed, directional by design; Table 8, Figure 7). That null is consistent with the sharpen-not-expand reading of RLVR and with documented spurious-reward effects on Qwen-lineage models [25, 18] – consistency, not confirmation, given the single seed.

What does not transfer. The effect sizes themselves. Every comparison in this paper lives between 42M and 1.19B training tokens on one 596M-parameter architecture at ≈2 tokens per parameter, far below compute-optimal [8]. These are fixed-budget attributions; their sign and size may change with budget, architecture, or data mixture, and the paper makes no scale claim (Section 13).

The cost of rigor. Controls and seeds multiply compute. Computed from the recorded per-cell budgets (Table 4), the de-confound ran 12 Phase-1 cells at 131,072,000 tokens each (1,572,864,000 tokens) plus nine Phase-2 cells at the same budget (1,179,648,000 tokens): ≈2.75B training tokens, roughly 2.3× the single 1.19B-token bundle run whose result they re-attributed. That multiple is the price of replacing "the bundle won" with "these components won, and these contributed

nothing." Nulls cost the same as wins: the SFT and RLVR stages spent full multi-arm budgets to establish that nothing separable happened (Table 7, Table 8).

What an evidence-gated loop buys. The gates run even when no one is looking for a specific error. None of the seven failures catalogued in Section 11 was caught by re-reading prose; all were caught mechanically – an implausibility check on a blend scoring roughly twice the bits-per-byte of both its parents (Table 5), a fixed held-out set that collapsed an in-loop 0.68-PPL "win" to +0.009 (Figure 6), and claim-to-file audits that corrected two already-recorded claims (a data-ratio pairing and a smoke-log misread) before writing. The same machinery that catches the errors validates the positive results; the discipline and the findings are one artifact.

13 Limitations

The gates of Section 3 bound what this study can claim; this section states the limits that remain, each with its consequence.

Scale and regime. Every controlled effect was measured on a single 596,049,920-parameter architecture with a single tokenizer (151,936-entry vocabulary), at budgets between ≈42M and ≈1.19B tokens (Table 3) – all far below the compute-optimal regime for this size [8]. The attribution cells are early-training proxies (2,000 steps, ≈131M tokens per cell), and effects measured there may grow, shrink, or invert over longer horizons: every result is a fixed-budget attribution, never a scaling claim. The descriptive gap-to-original numbers – a 2.14× perplexity gap at ≈30,000× less data (1.19B tokens) and 3.5× at ≈275,000× less data (131M tokens; Figure 2) – are in-loop validation perplexities, single run each, not suite-stamped, and carry no interval.

Evaluation. "Bit-exact" rests on a deterministic single-prompt fp32 forward comparison (Table 2); it verifies the forward computation exactly under that protocol and nothing broader. The same SmolLM2 weights run on GPU show logit deltas of 4.72×10^{-5} from kernel reduction-order differences (Table 2), so exactness is a CPU-fp32 statement. The code corpus noise floor is ≈68% of the measured value at the faithful baseline and reaches ≈92% across the three build checkpoints (text-lm-v2; Table 3), so code perplexity orderings are weakly powered and only the paired across-seed BPB intervals are relied on. BPB is measured on exactly two corpora – wikitext-2 and Python code, both English-adjacent – with no multilingual or broader-domain coverage. The SFT retention check on the standard suite used one representative seed per arm (Table 7); its 3-seed intervals apply only to the in-domain reasoning set.

Statistics. Across all cohorts, seeds vary initialization and shuffle order over a fixed data split, so the reported intervals likely understate full-randomization variance. The off-anchor AdamW learning-rate points are single-seed, weight decay was held at 0.1 for both optimizer arms rather than tuned per arm, and the sweep bounds AdamW only within 1.7e-3 to 4.8e-3 (Table 4). The GRPO cohort has n=1 seed and is capped at directional by design (math-acc-v1; Table 8). The passkey ladder uses n=40 probes per rung per cell, so individual rungs carry wide Wilson intervals (ecl-ladder-v1; Figure 5).

Unexplained observations. The base model's accuracy dip to 0.025 at its own 4,096-token trained window, repaired to 0.400 by both anneal arms (ecl-ladder-v1; Figure 5), has no defensible mechanism yet. The anneal's English improvement of +0.0157 BPB (95% CI [+0.0140, +0.0174],

3 seeds, paired, text-lm-v2; Table 6) violated its pre-registered expected null and is reported as an exploratory observation, not a claim.

Hardware. All training ran on one GB10 workstation with ≈119 GB of unified memory. The device's peak throughput is an estimate, so any MFU figure is approximate and never quoted as exact, and no result has been replicated on other hardware or in a distributed setting.

These limitations are precisely why several repository-level verdicts remain directional rather than win: the data-composition records are capped by conservative bookkeeping (Table 5), the SFT record by a disagreeing full-sequence cross-check (Table 7), and the RLVR record by its single seed (Table 8). Under this protocol, a limitation that cannot be ruled out becomes a cap on the verdict – which is the intended behavior.

14 Reproducibility Statement

All code, configurations, training and evaluation logs, decontamination reports, and figure-generation scripts are public at https://github.com/yashb98/BuildFromScratch. The entry point for any replication is the verification gate: `cd Qwen3-0.6B && python verify.py` builds the model from the single-file implementation, loads the published weights, and compares fp32 logits against the HuggingFace reference, passing only if the maximum absolute logit difference is below the 10^{-3} tolerance with next-token argmax agreement; the recorded result is exactly 0.0 (Table 2). The gate runs on CPU and needs no GPU.

Each experiment is isolated under `Qwen3-0.6B/experiments/<RUN_ID>/` and carries two machine-readable artifacts: `c5_evidence.json`, written before launch with the arm plan, seeds, token budgets, and the iso-FLOP confound check, and `verdict.json`, the scored outcome with per-seed values and confidence intervals; runs are also recorded in a project ledger written only through its programmatic interface (`research/ledger/ledger.py`). Every quantitative value in this paper traces to a named on-disk file: the evidence manifest that performs this mapping (`evidence_manifest.json`) ships with the paper source and lists, for each number, its source file and key or line, including the mandatory caveats attached to each value.

All training ran on a single NVIDIA GB10 (Grace Blackwell) workstation with ≈119 GB of unified CPU+GPU memory, using PyTorch with bfloat16 and `torch.compile`; the 151,936-way cross-entropy is computed in chunks, and every process caps its share of the unified pool before touching the accelerator. Multi-seed comparisons use seeds 0/1/2 (Table 1). Cross-run comparable numbers come only from versioned evaluation suites: text-lm-v2 (bits-per-byte and perplexity on pinned corpus revisions, wikitext-2-raw-v1 at commit `b08601e0` and codeparrot-clean-valid at commit `4db92d2e`), math-acc-v1 (a pinned answer extractor and verifier), and ecl-ladder-v1 (the passkey context ladder). In-loop trainer perplexities are labeled as such throughout and are never compared against suite-stamped numbers. Thirteen-gram overlap decontamination reports for the evaluation sets and prepared training sets are stored on disk beside the corresponding datasets and results.

15 Broader Impacts

This is a small-scale measurement study conducted entirely on public datasets: FineWeb-Edu [17], a DCLM-derived educational subset [11], OpenR1-Math, GSM8K [5], and MATH [7]. It releases no new model capability: the artifact is a 0.6B-scale reproduction of a published architecture [23]

trained on 1.19B tokens against the original's 36T (Figure 2), and every recipe component studied is already public.

The intended impact is methodological. Evidence-gated claims – multi-seed confidence intervals, iso-FLOP controls, versioned evaluation suites, and pre-registered gates – make small-model results cheaper to trust, and publishing controlled negative results reduces wasted replication compute for other resource-constrained groups. The SFT response-masking null (Table 7), the GRPO null at 0.6B (Table 8; pre-registered, and reported as directional under the study's own single-seed cap), and the context-extension stage gated off by a step-0 diagnostic are reported in full rather than left in the file drawer; the RLVR arms carry a content-blind random-reward control [18] so that the null is a measurement, not an absence of one.

The energy footprint is that of one desk-side workstation rather than a cluster: a single machine ran every experiment, wall-clock time is recorded per run in the shipped logs, and the largest single run – the 1.19B-token pretrain – completed in 2663.1 minutes (per-stage budgets in Table 1). Honest nulls at this scale are cheap to produce but rarely published; reporting them alongside the study's two attributed wins is the contribution this paper aims to make a norm.

16 Conclusion

This paper reproduced Qwen3-0.6B to bit-exactness ($\max|\Delta\text{logits}| = 0.0$ under the recorded single-prompt fp32 protocol; Table 2) and carried the reproduction through pre-training, data composition, mid-training, supervised fine-tuning, and RLVR under one evidence standard: single-variable diffs at matched train FLOPs, three paired seeds, bits-per-byte confidence intervals on two corpora, versioned eval suites, and pre-registration. Under that standard, an n=1 bundle headline decomposed into three small attributed architecture effects, an early-training optimizer-speed effect, and two null components (Table 4); data interventions produced the largest attributed effects (Tables 5 and 6); and three negative results – response masking failing to separate from its iso-FLOP control (Table 7), a pre-registered GRPO null reported as directional under the single-seed cap (Table 8), and context extension gated off at step 0 – are reported to the same standard as the wins, alongside a catalogue of the study's own corrected errors (Section 11). Two directions follow, stated as directions only: measuring whether these fixed-budget effects persist as the token budget grows toward compute-optimal, by re-running the same single-variable ladders at increasing budgets; and a multi-seed RLVR cohort, warranted only if a future base clears the Phase-1 solvability floor decisively. Neither direction contributes any result to this paper.

References

[1] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

[2] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

[5] 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, 2021. URL https://arxiv.org/abs/2110.14168.

[6] George Grigorev. Imu-1: Sample-efficient pre-training of small language models, 2026. URL https://arxiv.org/abs/2602.02522.

[7] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

[8] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

[9] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

[10] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies, 2024. URL https://arxiv.org/abs/2404.06395.

[11] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models, 2024. URL https://arxiv.org/abs/2406.11794.

[12] Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. Normuon: Making muon more efficient and scalable, 2025. URL https://arxiv.org/abs/2510.05491.

[13] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL https://arxiv.org/abs/1711.05101.

[15] Amirkeivan Mohtashami and Martin Jaggi. Landmark attention: Random-access infinite context length for transformers, 2023. URL https://arxiv.org/abs/2305.16300.

[16] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2024. URL https://arxiv.org/abs/2501.00656.

[17] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

[18] Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr, 2025. URL https://arxiv.org/abs/2506.10947.

[19] Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

[20] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021. URL https://arxiv.org/abs/2104.09864.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

[22] Sen Xu, Shixi Liu, Wei Wang, Jixin Min, Yingwei Dai, Zhibin Yin, Yirong Chen, Xin Zhou, and Junlin Zhang. Vibethinker-3b: Exploring the frontier of verifiable reasoning in small language models, 2026. URL https://arxiv.org/abs/2606.16140.

[23] 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.

[24] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

[25] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

[26] Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

Table 1: Per-stage training hyperparameters. Superscripts flag provenance: [R] reported in the source paper (Qwen3 technical report LR shape; VibeThinker SFT stage-1; GRPO/DAPO/Dr.GRPO), [I] inferred from a band in the source, [C] chosen by us (no published value at this scale). Shared unless stated: AdamW betas $(0.9, 0.95)$, eps 10^{-8}, grad-clip 1.0, bf16, seq. len. 4096, chunked fp32 cross-entropy, one GB10 GPU; all part-(a) stages use 65,536 tok/step (micro-batch 4 $\times$ grad-accum 4)[C]. Each run directory records its full recipe in `c5_evidence.json` before launch (contract C5); values are cross-checked against the `args` lines of the training logs. The optimizer A/B also ran a matched-config AdamW LR sweep $\{1.7, 2.4, 3.5, 4.8\} \times 10^{-3}$ (n=1 off-baseline). GRPO Phase 2 is n=1 by design (verdict capped at directional).

	Faithful 2TPP (06-08)	**Optimizer A/B** (06-16)	**De-confound P1/P2** (06-18 / 06-21)	**Data cells** (06-24 / 06-26)	**Mid-train anneal** (06-30)
Init	random	random	random	random	2TPP ckpt
Data	FineWeb-Edu[C]	FineWeb-Edu[C]	FineWeb-Edu[C]	dclm-edu; 50/50 mix[C]	dclm-edu, FineWeb[C]
Arms	—	AdamW vs NorMuon-2D	+WSD, +z-loss, +arch; VR, LNS, HG	dclm; mix (ctrl reused)	50/50 mix vs FineWeb-only
Optimizer	AdamW[R]	AdamW vs NorMuon[C]	AdamW[R]	AdamW[R]	AdamW[R]
LR peak $\rightarrow$ end	2.4e-3[C]$\rightarrow$ 3.2e-4[R]	2.4e-3[C]; NM 0.011[C] ($\rightarrow$ 0.1x)[C]	1.7e-3[R]$\rightarrow$ 3.2e-4[R]	1.7e-3[R]$\rightarrow$ 3.2e-4[R]	2.5e-4[C]$\rightarrow$ 2.5e-5[C]
Schedule	cosine[R]	cosine[R]	cosine[R] (WSD arm: WSD)	cosine[R]	1-sqrt decay[R] (shape)
Warmup (steps)	900[C]	50[C]	100[C]	100[C]	46[C]
Weight decay	0.01[R]	0.1 (2D)[C]	0.01[R]	0.01[R]	0.01[R]
Steps (tok)/cell	18,150 (1.19B, 2 TPP[C])	640 (41.9M[C])	2,000 (131M[C])	2,000 (131M[C])	2,300 (151M[C])
Cells (new)	1	6	12 / 9	3 / 3	6
Seeds	0	0–2	0–2	0–2	0–2

	SFT 3-seed (06-27)	**GRPO Phase 2** (07-02)
Init	faithful 2TPP base (strict load)	SFT seed-0 ckpt (policy + frozen KL ref)
Data	OpenR1-Math-220k pairs[C]	grpo-math-prompts-v1: GSM8K + MATH L1–3, 2:1 mix[C]
Arms	masked SFT vs no-mask iso-FLOP control	Dr.GRPO vs random-reward vs RFT
Optimizer / LR	AdamW, 5e-5[R]$\rightarrow$ 8e-8[R] cosine	AdamW, 1e-6 const.[I] (RFT: 1e-5[I])
Warmup / wd	5% (11 steps)[R] / 0.01[C]	none / 0.0[C]
Batch	524,288 tok/step (mb 4 x ga 32 = 128 seqs[R])	16 prompts x G=8 rollouts / step[C]
Steps (budget)	238 (125M tok[C])	300 (300 x 16 x 8 rollouts, cap 256 new tok[C])
Sampling	—	T = 0.8[C], no KV cache (exact re-forward)
Clip / KL	—	clip eps 0.2[R]; k3-KL beta 3e-3[I]
Reward	—	math-acc-v1: 1.0 / 0.1 / 0.0[C]; random arm Bernoulli(0.5)
Seeds	0–2	0 (n=1, directional)

Model	HF reference	Params	max $\lvert\Delta\text{logits}\rvert$	Argmax	Device
Qwen3-0.6B (ours)	`Qwen/Qwen3-0.6B-Base`	596,049,920	0.0	match	CPU/GPU fp32
SmolLM2-135M (ours)	`HuggingFaceTB/SmolLM2-135M`	134,515,008	0.0	match	CPU fp32
(same weights, GPU)			4.72×10^{-5}	match	GPU fp32

Table 2: Reproduction verification against the HuggingFace reference weights. Each check is a deterministic single-prompt fp32 forward-logits comparison (tolerance 10^{-3}; no training compute is compared, so no compute budget, seeds, or variance estimate applies, and no eval-suite stamp applies). Qwen3-0.6B: prompt "The capital of France is", identical next token (id 12095, " Paris"), `max_abs_error` and `relative_error` both exactly 0.0 (`verify.json`); the parameter count is the forward-calculation total asserted by the build's unit test (`architecture_plan.md`). SmolLM2-135M: bit-exact on CPU fp32 ($\max|\Delta| = 0.0$, identical next token " the", `parity.log/param_count.log`); the same weights on GPU show $\max|\Delta| = 4.72 \times 10^{-5}$, attributed to SDPA kernel-dispatch reduction-order noise, not a weight or architecture difference (`comparison_with_hf.md`).

	In-loop val PPL	wikitext-2 (text-lm-v2)		code_py (text-lm-v2)	
Build	(FineWeb-Edu, n=1)	PPL	floor	PPL	floor
Modernized (IMU-1)	23.52	27.80	1.07	129.42	111.92
Faithful (baseline)	28.65	37.01	1.30	438.67	280.56
Exploratory (partial-RoPE 0.25)	29.54	38.08	1.70	447.30	298.90

Table 3: The three Qwen3-0.6B builds at matched compute: each trained 18,150 steps × 65,536 tok = 1,189,478,400 (≈1.19B) FineWeb-Edu tokens at sequence length 4096 (identical data, schedule shape, and held-out val set). "In-loop val PPL" is the trainer's own FineWeb-Edu validation eval – a single run per build (n=1, no variance estimate, no suite stamp; the IMU-1 value is the last eval, at step 18,000 of 18,150). The right four columns are the independent `text-lm-v2` eval-harness pass on the same checkpoints (204,600 tokens per corpus) with its subsample noise floor (`floor_abs`); they confirm the ordering on both corpora, but the code_py floors are 68–92% of the measurement (floor_pct 92.1/68.2/70.9), so the code-PPL gaps should be read as directional only. A fourth arm (partial-RoPE 0.10) died at step 5,450/18,150 (last-eval PPL 50.71) and is excluded. Sources: `qwen3_imu1_2tpp_train.log`, `qwen3_baseline2tpp_train.log`, `qwen3_prope25_2tpp_train.log`, `phase_b_driver.log`, and the three per-build `suite_results.json` files.

		wikitext-2			code_py		
Study	Axis	ΔBPB	95% CI	sig.	ΔBPB	95% CI	sig.
Optimizer (42M tok)	NorMuon vs AdamW	+0.4743	[+0.4435, +0.5052]	yes	+0.5016	[+0.4560, +0.5471]	yes
Phase 1 (131M tok)	wsd	+0.0245	[−0.0173, +0.0662]	n.s.	+0.0361	[−0.0649, +0.1372]	n.s.
	zloss	−0.0034	[−0.0175, +0.0108]	n.s.	−0.0007	[−0.0522, +0.0508]	n.s.
	arch	+0.1179	[+0.1005, +0.1354]	yes	+0.3049	[+0.2588, +0.3511]	yes
Phase 2 (131M tok)	vr	+0.0355	[+0.0116, +0.0594]	yes	+0.1072	[+0.0492, +0.1651]	yes
	ln	+0.0337	[+0.0175, +0.0498]	yes	+0.0606	[+0.0198, +0.1013]	yes
	hg	+0.0256	[+0.0125, +0.0386]	yes	+0.0422	[+0.0081, +0.0763]	yes

Table 4: Attribution of the IMU-1 bundle via single-variable iso-FLOP ablations (ΔBPB > 0 means the treatment improves bits-per-byte over the faithful AdamW baseline; all numbers `text-lm-v2`-stamped, n=3 seeds per arm with 95% Welch-t CIs). Optimizer row: NorMuon vs AdamW on the 196 2D hidden weights only (embeddings/1D AdamW in both arms) at a fixed 41,943,040-token (≈42M) budget per cell, 640 steps × 65,536 tok, iso-FLOP by construction; a matched-config AdamW LR sweep (1.7/3.5/4.8e-3 single-seed plus the 3-seed 2.4e-3 anchor) spans only ∼0.047 BPB, ∼10x smaller than the gap, so no swept AdamW LR closes it; scoped as an early-training optimization-speed signal, not a scale claim. Phase 1/Phase 2 rows: 2000 steps × 65,536 tok = 131,072,000 tokens per cell against a shared 3-seed baseline (Phase 2 reuses the Phase-1 baseline checkpoints, re-scored in-cohort); iso-FLOP confirmed within the 5% gate (largest deviation: arch/hg +0.077% params, FLOP ratio 1.00043). Phase 1 finds *arch* the only significant driver (drivers = [arch], verdict "attributed"); Phase 2 splits arch into value-residual (vr), layernorm-scaling (ln), and head-gating (hg), each individually significant on both corpora (drivers = [vr, ln, hg]). Sources: `verdict.json` of the three experiments plus `lr_sweep_bpb.json`.

Table 5: Pretraining data composition at 596M: **dclm-edu** and a **50/50 dclm-edu + FineWeb-Edu mix** vs. a FineWeb-Edu control. Every cell trains the identical model for 2,000 steps × 65,536 tokens = 131,072,000 (≈131M) tokens (n=3 seeds per arm); only the data differs, so the arms are iso-FLOP by construction (train-FLOP ratio 1.000). ΔBPB = control BPB − arm BPB (positive = arm better); 95% CIs are paired across-seed; "sig." means the CI excludes zero. FineWeb-Edu control checkpoints are reused baselines (control-reuse policy); the mix experiment reuses the dclm arm. On code, the mix retains 0.5904/0.7034 = 83.9% of the pure-dclm improvement while directionally *improving* English (dclm alone directionally hurts it, n.s.). Eval suite: `text-lm-v2` (BPB with the model's own tokenizer; wikitext-2 val 869,710 bytes, Python code 843,643 bytes, 204,600 tokens per corpus per cell). Source files: `verdict.json`, `cohort_bpb.json`, `c5_evidence.json`, `run_arms.sh` (experiments 2026-06-24 dclm-vs-fineweb and 2026-06-26 data-mix-composition).

Arm	Corpus	Control BPB	Arm BPB	ΔBPB	95% CI	sig.
dclm-edu	code (Python)	2.6385	1.9351	+0.7034	[+0.6639, +0.7430]	yes
dclm-edu	wikitext-2	1.5157	1.5254	−0.0097	[−0.0255, +0.0061]	no
50/50 mix	code (Python)	2.6385	2.0481	+0.5904	[+0.5532, +0.6277]	yes
50/50 mix	wikitext-2	1.5157	1.4996	+0.0161	[−0.0005, +0.0327]	no

Table 6: Mid-training anneal at 596M: premium-mix anneal vs. an iso-token FineWeb-Edu control vs. the un-annealed base. Both arms run the identical 1-sqrt cooldown (peak LR $2.5 \times 10^{-4} \rightarrow 2.5 \times 10^{-5}$, 10% floor) for 2,300 steps × 65,536 tokens = 150.7M tokens per cell (≈13% of the 1.19B-token pretrain), n=3 seeds per arm; only the data differs (iso-FLOP by construction). Top block: BPB per corpus, with the anneal−control improvement and its paired across-seed 95% CI (suite `text-lm-v2`). The control barely moves off the base (+0.0050 code, +0.0036 wikitext-2 BPB), i.e. LR decay plus 150M extra FineWeb tokens alone do approximately nothing — the data, not the schedule, carries the effect. The wikitext-2 gain is small but significant, violating the pre-registered expected null on English; it is reported as-is. Bottom block: effective-context-length passkey ladder (suite `ecl-ladder-v1`, n=40 probes per rung per cell, arms pooled over 3 seeds, Wilson 95% CIs); the base's anomalous 4096-rung accuracy (exactly 0.025) is repaired to 0.400 by both arms, and no cell regresses vs. base at any rung. A separate step-0 diagnostic (`step0_diagnostic.json`) had gated context *extension* to propose-only: the base scores 0.083 at its own 4096-token trained window (threshold 0.5, gate FAIL). Source files: `verdict.json`, `c5_evidence.json` (2026-06-30 midtrain-anneal), `step0_diagnostic.json` (2026-06-27 midtraining).

	Base	Control (FineWeb)	Anneal (mix)	Anneal − Control (95% CI)
BPB (lower is better; improvements are positive), suite `text-lm-v2`				
Code (Python) BPB	2.1286	2.1236	1.8520	+0.2716 [+0.2641, +0.2792] sig.
wikitext-2 BPB	1.2256	1.2220	1.2062	+0.0157 [+0.0140, +0.0174] sig.
Δ vs. base, code	—	+0.0050	+0.2766	
Δ vs. base, wikitext-2	—	+0.0036	+0.0194	
Passkey accuracy by context rung (suite `ecl-ladder-v1`; Wilson 95% CI; arms pooled, 3×40 probes)				
512-token rung	0.200 [0.105, 0.348]	0.308 [0.233, 0.396]	0.817 [0.738, 0.876]	
4096-token rung	0.025 [0.004, 0.129]	0.400 [0.317, 0.489]	0.400 [0.317, 0.489]	

Table 7: SFT at 596M: response-masked SFT vs. an iso-FLOP `-no_mask` continued-pretrain control (identical data, LR schedule, and token budget; masking is the only variable) vs. the base. Recipe: ≈125M OpenR1-Math tokens (one pass, 125,000,592 prepared), peak LR 5×10^{-5} cosine $\rightarrow 8 \times 10^{-8}$, 5% warmup, n=3 seeds per arm. The in-loop comparison (top block) is **confounded — advisory only**: the SFT arm was scored on *response tokens only* while the control was scored on *all tokens*, so its 0.68-PPL "significant" gap is not a like-for-like number. Re-scored on one fixed held-out response-masked set (66 OpenR1-Math docs, 228,529 response tokens, 13-gram dedup vs. train), the gap collapses to +0.009 PPL; the exact tokenizer-free full-sequence cross-check flips sign (−0.006, control ahead, n.s. for improvement), so the masked and full-sequence comparisons disagree and the verdict of record is *directional* (not-yet-separable), while both arms improve ≈18% over base. Seed spread within each arm: SFT 0.001, control 0.004 PPL. Forgetting probe: held-out FineWeb-Edu PPL; wikitext-2/code retention is within the noise floor on the standard suite (`text-lm-v2`, representative seed). Source files: `verdict.json`, `reasoning_verdict.json`, `suite_results.json`, `c5_evidence.json` (2026-06-27 sft-3seed).

Evaluation (PPL, lower is better)	Base	SFT (n=3)	Control (n=3)	SFT − Control (95% CI)
In-loop advisory — CONFOUNDED: SFT scored on response tokens, control on all tokens				
In-loop train-window proxy	14.262	11.595	12.273	+0.679 [+0.676, +0.681] (not comparable)
Corrected: one fixed held-out response-masked set (verdict of record)				
Held-out masked reasoning PPL	14.127	11.573	11.582	+0.009 [+0.005, +0.014] sig.
Full-sequence cross-check	14.713	12.067	12.060	−0.006 [−0.011, −0.001] n.s. (control ahead)
Catastrophic-forgetting probe (held-out FineWeb-Edu; higher = more forgetting)				
FineWeb-Edu PPL	21.495	21.652	21.660	retained (+0.16 vs. base, ≈0.7%)

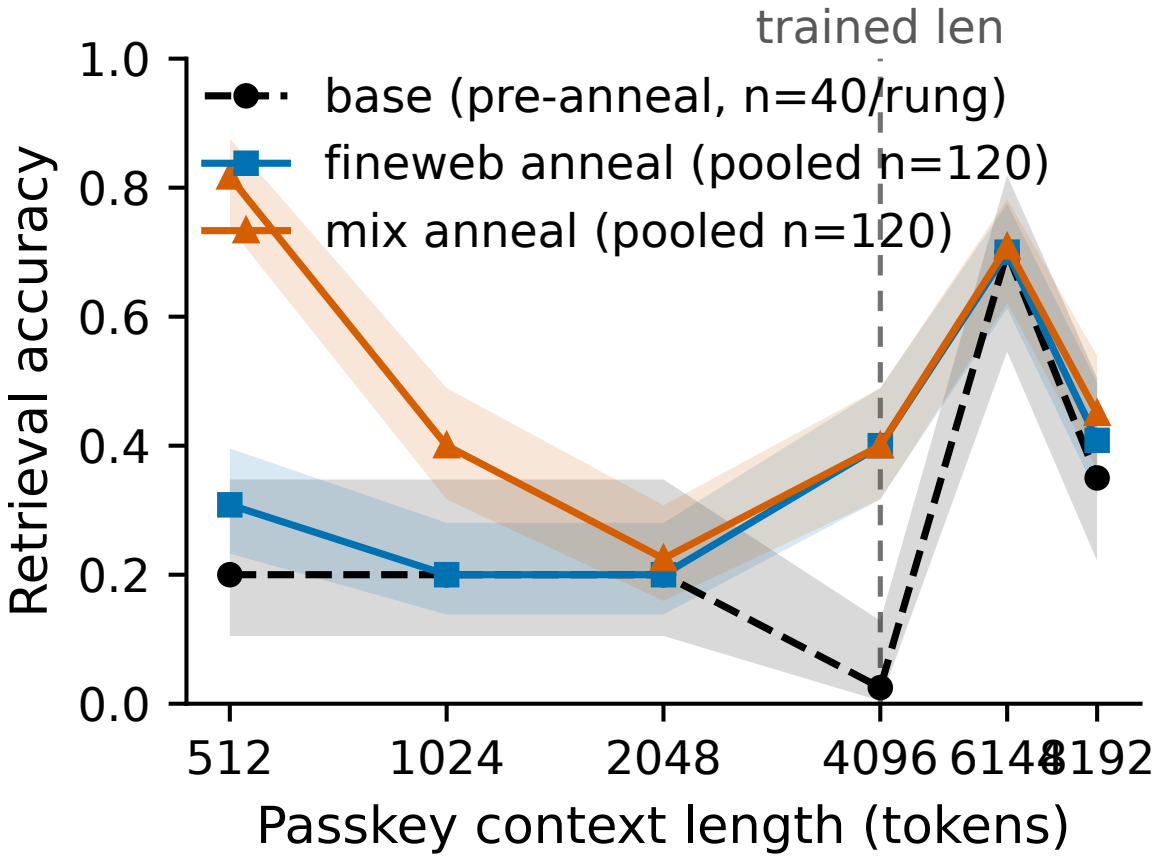

Figure 5: Effective-context-length passkey ladder (suite `ecl-ladder-v1`): accuracy by context rung (512–8192 tokens) for the un-annealed base, the FineWeb-Edu-anneal control, and the mix-anneal treatment. n=40 probes per rung per cell; anneal arms pooled over 3 seeds (3×40 probes per point); Wilson 95% CIs. Both anneal arms (150.7M tokens per cell, identical cooldown) repair the base's anomalous 4096-rung dip ($0.025 \rightarrow 0.400$), and no cell regresses versus the base at any rung. Sources: `ecl_ladder.json` and `verdict.json` (2026-06-30 midtrain-anneal); rendered by `fig_ecl_ladder.py`.

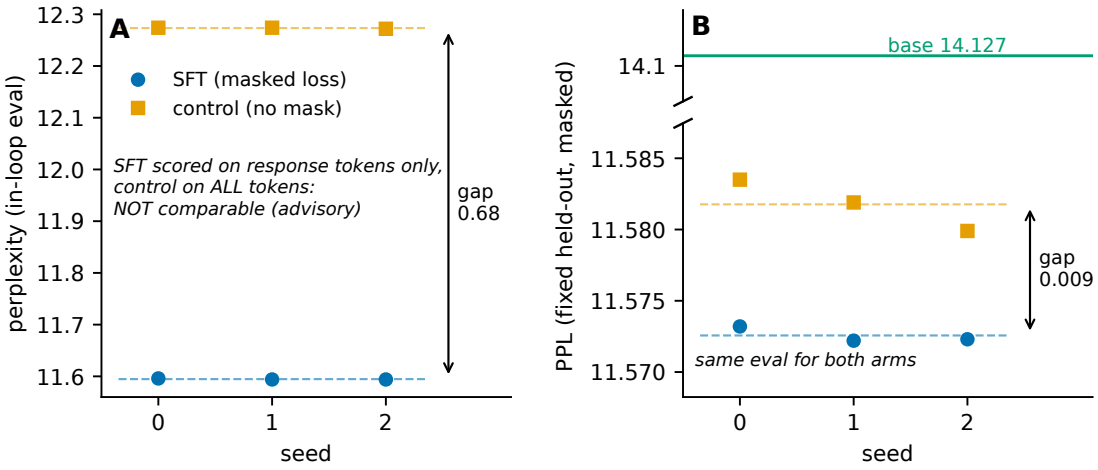

Figure 6: The SFT eval-token confound. Left: the in-loop advisory comparison (training-loop perplexity, not suite-stamped) scores the SFT arm on response tokens only and the control on all tokens, producing an incomparable +0.68-PPL apparent gap (SFT 11.595 vs. control 12.273). Right: re-scored on one fixed held-out response-masked set (66 OpenR1-Math documents, 228,529 response tokens, 13-gram dedup vs. train), the gap is +0.009 PPL (95% CI $[+0.0045, +0.0139]$) with a sign-flipping full-sequence cross-check (-0.006, n.s.) — verdict directional. Budget: one pass over 125,000,592 OpenR1-Math tokens per arm, iso-FLOP, masking the only variable; n=3 seeds per arm; across-seed spread SFT 0.001 / control 0.004 PPL. Source files: `verdict.json`, `reasoning_verdict.json` (2026-06-27 sft-3seed); rendered by `fig_sft_confound.py`.

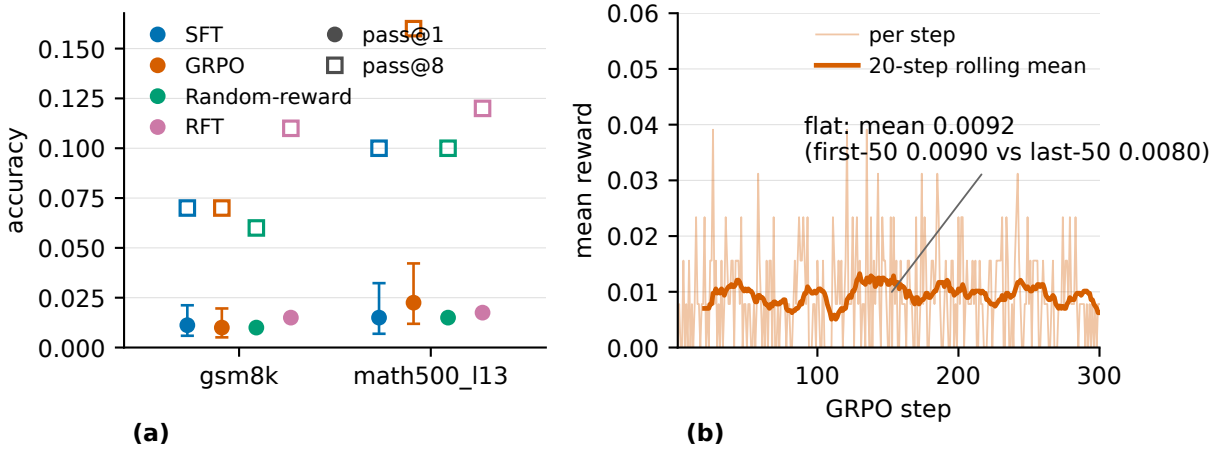

Figure 7: RLVR at 596M: pass@1 (Wilson 95% CIs) and pass@8 [3] for the SFT floor, Dr.GRPO, the random-reward gate arm, and the iso-generation-compute RFT control on GSM8K (100 items) and MATH-500 levels 1–3 (50 items), 8 samples per item at T=0.8, band seed 20260701, n=1 training seed, verifier pinned as `math-acc-v1`. GRPO trained 300 steps × 16 prompts × G=8 rollouts; its training reward stayed flat at ≈0.9% verifier-correct with no trend. All arm CIs overlap and both pre-registered gates fail. Source files: `phase1_passk.json`, `passk_grpo.json`, `passk_random.json`, `passk_rft.json`, `health_grpo_seed0.jsonl`.

Table 8: RLVR at 596M: paired pass@k of the SFT floor, Dr.GRPO, the random-reward (Spurious-Rewards) gate arm, and the iso-generation-compute RFT control, all decoded on the same band (100 GSM8K + 50 MATH-500 level-1–3 items, 8 samples per item, T=0.8, max 256 new tokens, band seed 20260701; answer verifier pinned as `math-acc-v1`; both eval sets 0-flagged in a 13-gram decontamination check against the 35,924 SFT problems). GRPO trained 300 steps × 16 prompts × G=8 rollouts; its training reward stayed flat at ≈0.9% verifier-correct with no trend. The RFT control collected 352 verifier-correct completions (≈0.9% of 38,400 rollouts) — not 0, as an earlier draft misread from a smoke-test log line. pass@1 carries a Wilson 95% CI; pass@8 is the Chen et al. (2021) unbiased estimator; a gate "passes" only on non-overlapping Wilson CIs. Both pre-registered gates FAIL on both sets, confirming the pre-registered null; RFT is nominally best on GSM8K (pass@8 11% vs. 7%) and GRPO on MATH-500 (16% vs. 10%), but all CIs overlap (n.s.), and GRPO's MATH-500 pass@1 merely ties the pretrain base (2.25%). n=1 seed, so the verdict is capped at *directional*; phase 3 not launched. Source files: `phase1_passk.json` (SFT, base), `passk_grpo.json`, `passk_random.json`, `passk_rft.json`, `verdict.json`.

	GSM8K (100 items)		MATH-500 L1–3 (50 items)	
Arm	pass@1 % (Wilson 95%)	pass@8 %	pass@1 % (Wilson 95%)	pass@8 %
SFT (floor)	1.12 [0.59, 2.12]	7	1.50 [0.69, 3.23]	10
GRPO	1.00 [0.51, 1.96]	7	2.25 [1.19, 4.22]	16
Random reward (gate)	1.00 [0.51, 1.96]	6	1.50 [0.69, 3.23]	10
RFT (iso-gen. control)	1.50 [0.86, 2.60]	11	1.75 [0.85, 3.57]	12
Base (reference, Phase 1)	0.62 [0.27, 1.45]	5	2.25 [1.19, 4.22]	14
Pre-registered gates (non-overlapping Wilson CIs required; outcome identical on both sets)				
GRPO beats SFT floor	FAIL (`grpo_beats_sft_floor` = false)			
GRPO beats random gate	FAIL (`grpo_beats_random_gate` = false)			