Marin: Fully Open LLM Training at 8B and 32B with Mid-Flight Adaptation

marin team

February 2026

Abstract

Fully open language-model releases enable reproduction and scientific study beyond what is possible from open weights alone. We present a retrospective of MARIN, a community-driven effort that releases not only checkpoints, but also training code, data mixtures, intermediate artifacts, and an issue-driven development history. We document two base-model runs: Marin 8B, trained through multiple cooldown/reheat phases with data-mixture evolution and deep-cooldown stability interventions; and Marin 32B, which required a mid-run switch to QK-norm attention to eliminate loss spikes and a shuffling redesign (Feistel permutation) to avoid late-training phase-shift pathologies. Beyond final benchmark numbers, we emphasize actionable lessons about stability tooling, data-mixture iteration (microannealing), shuffle correctness, and contamination hygiene.

1 Introduction

The open-weight LLM ecosystem has grown rapidly, with strong releases at many sizes (e.g., Llama [10], Qwen [31, 41], Gemma [9]). However, model weights alone are rarely sufficient to reproduce training dynamics, diagnose failures, or study how data and optimization choices shape capabilities. A complementary line of work has advocated "fully open" releases: not just final checkpoints, but also training code, data mixtures, intermediate artifacts, and documentation that enables reproduction and scientific inspection [2, 11, 19, 20, 29].

In this paper we document MARIN, a community-driven, fully open effort to train competitive base language models with transparent "model flow" artifacts (data mixes, code, checkpoints, and a public issue-driven development history) [13]. Our goal is not only to report final scores, but to provide a phase-by-phase retrospective of what actually happened in long, expensive training runs: what broke, what interventions worked, and which fixes were "one-way doors" versus easily reversible.

What are "8B" and "32B"? They denote the approximate parameter count of the two primary Marin base models. The 8B run targets the widely used single-node or modest-cluster deployment regime, while the 32B run targets a higher-capability regime comparable to recent open-weight 30–32B baselines.

What are the "issues" in the retrospectives? They are concrete operational and scientific problems encountered at scale: loss spikes, attention instability, batch/LR transitions, data-mixture regressions, shuffling pathologies, benchmark contamination, and implementation mistakes. These issues are often under-emphasized in final model cards, but are central to making training reliable and reproducible.

1.1 Contributions

We make four contributions.

1. **A fully open training record for Marin 8B and 32B.** We summarize architectures, data mixtures, hyperparameters, phase boundaries, and evaluation settings, and we point to public experiment specifications and issue threads that motivated major interventions [13].

2. **A pragmatic "mid-flight adaptation" methodology.** We show how a single long run can incorporate multiple cooldowns, reheats, and data-mixture changes ("tootsie roll" training), guided by frequent evaluation and low-cost "microannealing" experiments (cf. midtraining and micro-annealing in OLMo 2 [29]).

3. **Stability lessons at two scales.** For Marin 8B, we highlight how EMA monitoring, deep cooldowns, and z-loss became important as learning rates approached very low regimes. For Marin 32B, we document how "optimizer-side" mitigations softened but did not remove loss spikes, and how a mid-run switch to QK-norm attention eliminated spikes after a short recovery window [8].

4. **A case study in data-system failures and fixes.** We describe how shuffling quality can produce late-training phase shifts, motivating a switch from an affine/LCG permutation to a Feistel-network permutation. We also document a benchmark contamination incident (GSM8K) introduced via cached data, and the safeguards added afterward.

1.2 Paper Roadmap

Section 2 surveys related work. Section 3 details the Marin pipeline, data, architectures, training schedules, and artifact release. Section 4 reports the 8B and 32B retrospectives, including figures reproduced from the official retrospectives and benchmark results. Section 5 concludes with limitations and future directions.

2 Related Work

2.1 Open-Weight LLMs

Recent open-weight technical reports provide strong baselines and highlight engineering tradeoffs in training large models. The Llama family [10] helped popularize detailed reporting and the idea of using mid-training interventions. Qwen 2.5 and Qwen 3 [31, 41] and Gemma 3 [9] provide competitive 27–32B class baselines. While these reports provide substantial implementation details, they typically do not release full training data mixtures and intermediate artifacts.

2.2 Fully Open Training Releases

A smaller set of projects release not only weights, but also training code and dataset compositions, enabling reproduction and deeper scientific study. Examples include OLMo [11] and OLMo 2 [29], Pythia [2], LLM360 [20], and DataComp-LM/DCLM [19]. Marin is aligned with this "fully open" ethos, emphasizing a public, issue-driven record of experimentation and the release of artifacts along the training lifecycle [13].

2.3 Data Curation and Web-Scale Corpora

Open corpora and documented data pipelines are increasingly central to reproducible LLM research. Dolma provides an open multi-trillion-token corpus and standardized subsets for experimentation [33]. Nemotron-CC offers a refined Common Crawl-derived corpus designed for long-horizon pretraining [34]. For code, StarCoder/Stack-style corpora provide broad coverage of permissively licensed repositories [22]. Marin's retrospectives highlight that "high-quality" sources (e.g., Wikipedia, ArXiv) may be missing structures that matter for few-shot benchmarks; mixing in instruction-like collections such as FLAN can partially counteract this [38].

2.4 Multi-Stage Training and Midtraining

Chaining multiple training stages is a common strategy to patch deficits, add domains, or increase "post-trainability." This includes continued pretraining in new domains [12] and more explicit midtraining stages described in recent reports [1, 29, 30]. End-of-training domain upsampling can yield gains on target capabilities, but may introduce regressions elsewhere [3]. Marin adopts a practical version of this idea: cooldown phases with targeted data mixes, plus short, low-cost "microannealing" runs to evaluate candidate sources before committing substantial compute.

2.5 Training Stability and Optimization

Large runs often exhibit loss spikes and other instabilities that can waste compute and degrade final performance. Stability toolkits include architectural interventions such as QK-norm [8], regularizers such as z-loss [4, 6, 40], and optimizer-side heuristics such as gradient clipping and skipping outlier steps. OLMo 2 provides an extended case study of instability diagnosis and mitigations (e.g., embedding norm dynamics, spike scoring, and skip-step optimizers) [29, 36]. Marin's 32B run reinforces a similar conclusion: heuristics can reduce severity but may not remove spikes at scale, motivating more fundamental attention-stack changes.

2.6 Post-Training

While this paper focuses on base-model development, Marin's 8B retrospective includes a small supervised fine-tuning (SFT) experiment to probe "SFT-ability." Open post-training recipes such as Tulu 3 [17] provide an end-to-end alignment pipeline (SFT, preference tuning, and RL variants). Marin's SFT results echo observations in OLMo 2 that instruction tuning can improve instruction-following metrics while degrading some base-model benchmarks, motivating continued pretraining mixing or more careful SFT data design [29].

3 Methods and Open Artifacts

This section summarizes the Marin "model flow" components that are necessary for reproduction: open artifacts, data sources and mixes, model architectures, and training/evaluation setup. Where possible we cite public specifications and the official retrospectives for exact numbers and plots [13, 24, 25].

3.1 Fully Open Artifacts (Model Flow)

Marin aims to be more than an "open weights" release. For each major run, the project publishes: (i) training code and experiment specifications, (ii) explicit data mixture manifests (dataset IDs and

weights), (iii) intermediate checkpoints, and (iv) a public issue-driven record of experiments and regressions. This framing is aligned with fully open releases such as OLMo/OLMo 2 [11, 29].

3.2 Data Sources

Marin's base-model runs draw from widely used open corpora:

- **DCLM Baseline and DCLM HQ** [19]: used heavily in the early 8B phases.
- **Dolma subsets** [33]: Wikipedia, StackExchange, ArXiv, and other web-derived sources; used in the 8B cooldown mixture.
- **Nemotron-CC** [34]: a refined Common Crawl corpus; used as the backbone of the 8B Phoenix/Starling phases and the 32B pretraining mix.
- **Code data (StarCoder/Stack)** [22]: used throughout to preserve coding capability.
- **Math-focused corpora**: FineMath-3+ and curated math bundles used in cooldowns, including Dolmino math (which surfaced a cached GSM8K contamination incident) and later MegaMath and Common Pile EDU-filtered Python [16, 24, 42].
- **New Marin datasets**: Markdownified corpora (ArXiv/StackExchange/Wikipedia) and Marin Datashop Science QA introduced in the 8B Starling cooldown.

Why format diversity matters. The 8B retrospective reports that changing data formatting (e.g., trailing whitespace conventions and Markdown structure) can move evaluation perplexity substantially (e.g., Paloma `c4_en`), suggesting that format diversity is a meaningful axis of distribution shift even when benchmark accuracy differences are small [23, 25].

3.3 Model Architectures

Marin 8B. Marin 8B uses a Llama-style decoder-only Transformer implemented in Levanter [25]. The run standardizes on sequence length 4096 and the Llama 3 tokenizer.

Marin 32B. Marin 32B begins with a Llama-3-style 32B configuration and later switches to a Qwen3-style attention stack that adds QK-norm [8, 24, 41]. The key outcome is that QK-norm provided headroom against loss spikes at 32B, while warm-starting preserved progress in embeddings/MLPs.

3.4 Optimization, Schedules, and Stability Tooling

Optimizer. Both 8B and 32B runs use AdamW [21]. For 8B, the retrospective reports mixed precision (parameters and optimizer states in float32, compute in bfloat16) and no weight decay on embeddings or layer norms [25].

Learning-rate schedules: WSD-S and WSD. The 8B run begins with a cyclic warmup-stable-decay schedule (WSD-S) and later switches to a more standard warmup-stable-decay schedule (WSD) during and after major transitions [25, 39]. WSD-S enables periodic cooldown probes during a long high-LR plateau, providing more frequent signals for intervention.

EMA monitoring. During 8B Phase 2, Marin adds an exponential moving average (EMA) of weights for monitoring evaluation loss. The retrospective highlights a surprisingly stable "EMA gap" (difference between hot and EMA eval loss) during high learning rates [25].

z-loss for deep cooldowns. While z-loss is commonly used as a stability regularizer in large-scale training [4, 6, 40], Marin's 8B retrospective emphasizes its practical value during deep cooldowns: adding a z-loss term prevented an `lm_head` norm blow-up observed in the Spoonbill cooldown [25].

3.5 Hardware, Attention Kernels, and Checkpointing

Marin training runs use TPU hardware (TPU Research Cloud). The 8B run uses 2x v5e-256 slices coordinated via multislice in Phase 1 and a v4-2048 slice thereafter, using JAX Splash Attention [25]. The 32B run begins on preemptible v5p-512 multislices and later migrates to a reserved v4-2048 slice, with a batch schedule adjusted for divisibility across slice count [24].

The 8B run saves permanent full checkpoints every 20k steps, with more frequent temporary checkpoints pruned over time [25].

3.6 Shuffling and Sampling Permutations

Motivation. At multi-trillion-token scales, it is not enough for the training order to be a bijection over indices; it must also mix well locally so that contiguous steps see approximately i.i.d. batches. The 32B retrospective motivates this as reducing within-batch correlation (avoiding long correlated stretches that can bias updates) and reducing gradient variance from batch to batch. The Marin 32B cooldown surfaced a concrete failure mode: training loss "phase-shifted" late in cooldown while validation remained stable, indicating a data-ordering artifact rather than model divergence [24].

Stateless PRP shuffling in Levanter. Marin trains with Levanter's deterministic, resume-friendly data pipeline, which supports applying a pseudo-random permutation (PRP) to dataset indices inside the loader rather than materializing a full shuffle table. In the Levanter implementation used by Marin, the permutation is computed on-the-fly from a small set of parameters (keys), enabling random access and exact reproducibility across preemption/resume [18]. Concretely, Levanter exposes a permutation type switch ("linear" vs "feistel") so that experiments can change mixing behavior without changing the underlying datasets [18].

Linear/LCG (affine) permutation. The original permutation used in the 32B run was an affine map modulo the dataset length N:

$$p(x) = (ax + b) \bmod N, \qquad \gcd(a, N) = 1, \tag{1}$$

with a and b sampled once per dataset from a PRNG seed [18, 24]. Equation 1 is a valid permutation because a is invertible modulo N, and it is extremely cheap: each index requires only a multiply, add, and modulo. However, the induced visit order $p(0), p(1), \ldots$ is an arithmetic progression on the ring $\mathbb{Z}_N$ (a fixed "stride" a and offset b). In particular, adjacent positions are always separated by the same modular distance, $p(x+1) - p(x) \equiv a \pmod N$, so the mapping provides no local "avalanche" behavior. If the underlying dataset is stored with structure (e.g., blocks grouped by source, shard, or preprocessing epoch) and is not itself pre-shuffled, a single-stride walk can yield long correlated stretches. Operationally this can manifest as non-stationary "phases" in training loss even when mixture weights over datasets remain constant [24].

Feistel permutation. In the Mantis cooldown, Marin switched to a Feistel-network PRP, which mixes the bit representation of indices through multiple rounds. Levanter's Feistel permutation embeds the domain $[0, N)$ into $[0, m)$ where $m = 2^{\lceil \log_2 N \rceil}$, splits the $\log_2 m$ bits into left/right halves, and applies several Feistel rounds with per-round keys; for non power-of-two N it uses cycle-walking (reapplying the network until the output falls back in $[0, N)$) to preserve bijectivity [18]. In a standard Feistel construction, each round updates (L, R) as $(R, L \oplus F(R, k_i))$, so information from the right half is repeatedly mixed into the left and vice versa. Concretely, Levanter uses $r = 5$ rounds by default and a simple round function over the right half, $F(R, k) = ((R \cdot 2654435761) + k) \bmod 2^{|R|}$, which is sufficient to destroy the linear structure of Equation 1 while retaining the same stateless/random-access property [18]. Empirically, this switch removed the cooldown phase shift and improved validation losses (including Paloma corpus-fit metrics) [24].

Takeaway. Marin's experience suggests treating shuffle quality as a testable systems component: a permutation can be correct (a bijection) and still be a poor shuffle when the data source itself is structured.

4 Experiments and Retrospectives

This section presents the Marin 8B and 32B retrospectives as a set of empirical case studies. We include key plots from the official retrospectives (downloaded into 'figures/') and summarize what each intervention changed [24, 25].

4.1 Evaluation Setup

Harness. Both retrospectives report results using EleutherAI's LM Evaluation Harness with task-default settings, which can differ from numbers reported in other frameworks due to prompt/template and strictness differences [24, 25].

Benchmarks. We report standard academic and reasoning suites. For base models, we emphasize MMLU [14], GSM8K [7], MATH [15], HumanEval [5], BBH [35], GPQA [32], and MMLU-Pro [37]. We also track corpus-fit metrics such as Paloma [23].

Contamination caveat. Both the 8B and 32B retrospectives emphasize that many popular benchmarks are contaminated in modern pretraining corpora. Marin additionally encountered a concrete contamination incident (GSM8K) due to cached data during a 32B cooldown, motivating stricter dataset-content checks going forward [24].

4.2 Marin 8B Retrospective

The Marin 8B run is a single long training trajectory partitioned into phases after the fact. The run used a Llama-style architecture in Levanter, sequence length 4096, JAX Splash Attention on TPUs, and mixed float32/bfloat16 precision [25].

4.2.1 Phase 1: Kestrel (DCLM + WSD-S)

Scope. Kestrel covers the first ~2.7T tokens of the 8B run (0→2.7T), trained on a reserved 2x v5e-256 TPU setup under a DCLM-centric mixture and a cyclic WSD-S schedule [25].

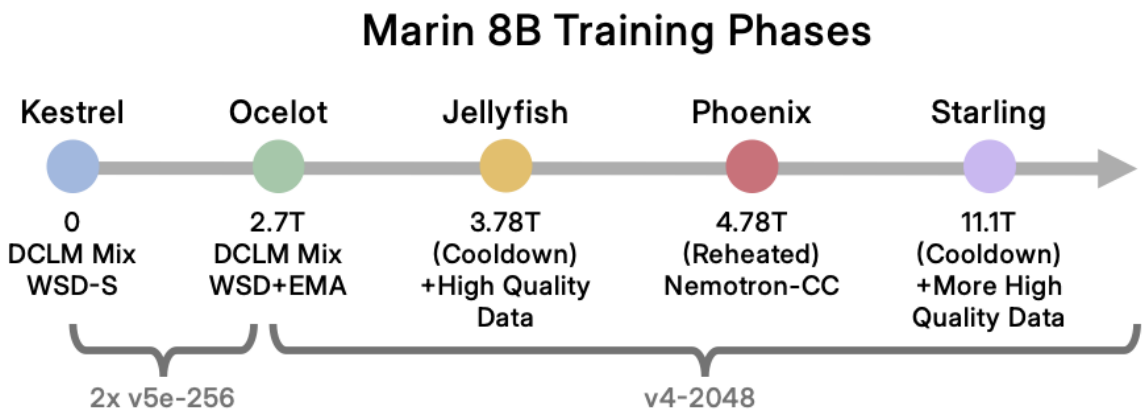


Figure 1: Phase partitioning for the Marin 8B run (reproduced from the retrospective) [25].

Dataset	Share
DCLM Baseline	92.6%
StarCoder Data	6.1%
ProofPile 2	1.3%

Table 1: Marin 8B Phase 1 (Kestrel) data mix (normalized shares) [25].

Data. Kestrel trains from scratch on the DCLM "best mixture" of DCLM Baseline, StarCoder Data, and ProofPile 2 [19, 25].

Schedule. Kestrel uses a cyclic warmup-stable-decay schedule (WSD-S) to enable periodic cooldown probes without pre-registering a single final decay [25, 39]. The retrospective reports that increasing the spacing of decay cycles (fewer, longer decays) improved several evaluation losses while revealing surprising domain-dependent behavior due to preprocessing mismatches (e.g., trailing whitespace conventions across Paloma subsets). Concretely, the run began with decays every 10k steps for 1k steps, then switched around step ~200k to decays every 20k steps for 2k steps (keeping the overall decay fraction similar) [25].

4.2.2 Phase 2: Ocelot (Batch/LR scaling + EMA)

At ~2.7T tokens the run moved from 2x v5e-256 to a v4-2048 slice. To better utilize the hardware, batch size was increased to 12Mi tokens/step and the learning rate was scaled by $\sqrt{3}$ to 1.7e-3 following batch-scaling heuristics [25]. During this phase Marin switched from WSD-S to WSD and introduced EMA monitoring of evaluation loss (EMA $\beta = 0.995$), holding the learning rate high through ~3.78T tokens [25]. A notable operational fix was correcting rotary embedding settings (Llama 2 to Llama 3 style), which the authors believe caused a brief loss spike.

The retrospective reports a representative EMA gap of ~0.015 bits-per-byte on Paloma `c4_en` at hot learning rates, with the gap changing primarily when the learning rate regime changes rather than trending over time [25].

Dataset	Share
Dolmino DCLM HQ	67.8%
Dolma peS2o	10.8%
FineMath 3+	6.3%
Dolma ArXiv	5.2%
Dolma StackExchange	3.2%
StarCoder	2.2%
Dolma Algebraic Stack	2.1%
Dolma Open Web Math	0.9%
Dolma Megawika	0.8%
Dolma Wikipedia	0.7%

Table 2: Marin 8B Phase 3 (Jellyfish) data mix (normalized shares) [25].

4.2.3 Interlude: Microannealing

Marin ran "microannealing" experiments: short cooldown-like runs that replace a fraction of the pretraining mix with a candidate "high-quality" source to estimate downstream impact cheaply. Consistent with prior observations on midtraining [29], naive HQ oversampling improved HQ-domain losses but degraded benchmark accuracy. The retrospective argues that HQ sources often lack task-like structures useful for few-shot accuracy, and that mixing in FLAN mitigates this effect; the best microannealing results came from 70% PT / 15% FLAN / 15% HQ [25, 38].

4.2.4 Phase 3: Jellyfish (First cooldown)

At ~3.78T tokens, DCLM tokens were running low, motivating a cooldown with a higher-quality mixture and a linear decay from 1.7e-3 to 1.7e-4 over 1e12 tokens (79,500 steps at 12Mi tokens/step) [25]. In the retrospective phase partitioning, Jellyfish spans ~3.78T→4.78T tokens [25].

The Jellyfish checkpoint achieved MMLU 5-shot 65.3 and GSM8K 8-shot 50.9, competitive with contemporary 7–8B open baselines, but still lagging on instruction-following metrics [25]. The retrospective notes that Paloma `c4_en` perplexity increased substantially under this mix, likely due to formatting differences between DCLM HQ and `c4_en`.

4.2.5 Phase 4: Phoenix (Reheat + Nemotron-CC transition)

After the first cooldown, at ~4.78T tokens, the run "reheated" and transitioned from the DCLM-based mix to Nemotron-CC plus StarCoder. The transition used a 2,000-step interpolation period (~25.2B tokens) with mixture weights proportional to token count; the learning rate was rewarmed linearly from 1.7e-4 to 1.7e-3 over the same window and then held fixed [25].

4.2.6 Deeper cooldowns: Raccoon and Spoonbill (z-loss)

To improve "SFT-ability," Marin ran deeper cooldown experiments while Phoenix continued. Raccoon cooled the Jellyfish checkpoint further to 1.7e-5 over ~100B tokens and observed an unexpected slow increase in training loss. Spoonbill reproduced the phenomenon and isolated an `lm_head` norm explosion; adding z-loss with weight 1e-4 stabilized training and removed the loss creep [25].

Dataset	Proportion	Oversampling
Nemotron CC Medium	22.1%	1x
Nemotron CC HQ Synth	17.8%	1x
Nemotron CC Medium Low	10.1%	1x
Nemotron CC HQ Actual	6.0%	1x
Nemotron CC Medium High	5.4%	1x
Nemotron CC Low Actual	4.6%	1x
Nemotron CC Low Synth	4.1%	1x
Marin ArXiv Markdown	5.2%	5x
Dolmino peS2o	5.2%	5x
StarCoder Data	4.5%	1x
ProofPile 2	4.5%	1x
FineMath 3+	3.0%	5x
Dolmino FLAN	3.0%	10x
Dolmino StackExchange	1.5%	5x
Marin StackExchange Markdown	1.5%	5x
Dolmino Math	0.8%	10x
Marin Wikipedia Markdown	0.3%	5x
Dolmino Wiki	0.3%	5x
Marin Datashop Science QA	0.1%	5x

Table 3: Marin 8B Phase 5 (Starling) cooldown mix, as reported in the retrospective [25].

Model	Avg	MMLU (5)	BBH	GPQA	MMLU-Pro	GSM8K
Marin 8B Base (Deeper Starling)	66.6	67.6	50.6	30.3	36.5	61.3
Llama 3.1 Base (8B)	65.3	66.4	46.4	32.3	33.3	56.8
OLMo 2 Base (7B)	64.9	63.9	44.4	26.8	30.6	67.6

Table 4: Selected base-model results reported in the Marin 8B retrospective (LM Eval Harness defaults) [25]. "MMLU (5)" denotes 5-shot.

4.2.7 Phase 5: Starling (Second cooldown + new datasets)

At ~11.1T tokens, Marin began a second cooldown, incorporating lessons from Raccoon/Spoonbill: deeper decay to 1.7e-5, z-loss 1e-4, and a batch increase to 16Mi tokens/step. This cooldown ran for 1.34T tokens (80k steps) [25]. The mix was approximately 70% Nemotron-CC and 30% high-quality sources, including new Markdownified datasets and a science-QA dataset.

The retrospective reports that `c4_en` perplexity decreased substantially during Starling, in contrast to the earlier cooldown where it increased, consistent with large preprocessing/formatting shifts between DCLM HQ and Nemotron-CC.

4.2.8 Base-model benchmark results (8B)

Table 4 summarizes key benchmark results from the 8B retrospective for the Deeper Starling checkpoint. The retrospective emphasizes that many tasks are contaminated in modern pretraining corpora and that these comparisons should be treated cautiously [25].

Model	Avg	IFEval	BBH	GPQA	MMLU-Pro	HumanEval
Llama 3.1 Tulu	50.0	87.5	43.9	28.7	29.4	60.4
Marin 8B SFT	43.8	78.3	46.0	29.5	31.2	47.0
OLMo 2 Instruct	38.7	69.5	42.6	24.2	17.6	17.1

Table 5: Selected SFT/instruct results reported in the Marin 8B retrospective (Open LLM Leaderboard hard set + additional tasks) [25].

Dataset	Share
nemotron_cc/medium	30.69%
nemotron_cc/hq_synth	24.70%
nemotron_cc/medium_low	13.98%
nemotron_cc/hq_actual	8.30%
nemotron_cc/medium_high	7.49%
nemotron_cc/low_actual	6.37%
nemotron_cc/low_synth	5.70%
starcoderdata	2.27%
proofpile_2	0.50%

Table 6: Marin 32B pretraining mixture (normalized share), reused across Phases 1–3 [24].

4.3 Marin 8B Supervised Fine-Tuning (SFT)

To probe post-trainability, the retrospective reports a small SFT run starting from the final Deeper Starling checkpoint. The SFT mix combines multiple open instruction/reasoning datasets (AceCode-89K, Bespoke-Stratos-17k, dolphin-r1, natural_reasoning, OpenThoughts, smoltalk, tulu-3-sft-mixture, verifiable-math-problems) and trains for 5Gi tokens with batch size 512Ki and learning rate 1.7e-4 [25].

The reported results (Table 5) show substantial gains on instruction-following and reasoning suites, but a degradation on some "base" tasks, echoing a phenomenon reported in OLMo 2. The retrospective proposes mitigation via mixing in pretraining data and FLAN during later stages [25, 29].

4.4 Marin 32B Retrospective

The Marin 32B run scales the 8B recipe to a 32B configuration and surfaces two main challenges: (i) training instability (loss spikes) during baseline Llama-style attention, and (ii) data-system pathologies during cooldown (benchmark contamination and shuffling). The final released artifact includes ~6.437T tokens of training (Phase 1 + Phase 3 + Phase 4; Phase 2 diagnostic restarts excluded) [24].

4.4.1 Phase 1: Baseline scale-up and loss spikes

Phase 1 trains a Llama-3-style 32B model for ~2.679T tokens (80k steps at sequence length 4096) using the Nemotron-CC-centric mixture in Table 6 and the batch schedule in Table 7 [24]. The baseline optimizer is AdamW with peak learning rate 7e-4 (WSD-style linear warmup/hold/decay; warmup 1% of steps, decay 40%), weight decay 0.05, EMA $\beta = 0.995$, and z-loss 1e-4 [24]. Loss spikes were frequent.

Start step	Global batch	Tokens/batch
0	8192	32Mi
18,500	7680	30Mi
21,010	8192	32Mi

Table 7: Marin 32B Phase 1 batch schedule (4096 sequence length) [24].

The retrospective reports three progressively stronger mitigations: (1) tightening max grad-norm clipping from 1.0 to 0.2 at ~56.4k steps (typical norms were ~0.2, and larger norms often preceded spikes), (2) adding "clip update norm" at step 72,233 using a rolling window of 128 updates and a 2σ threshold (briefly disabled around ~74k–80k), and (3) enabling "skip bad steps" to skip parameter updates whose update norm exceeds the same 2σ criterion [24].

Issue #1368 adds fine-grained observations: Adam update-norm spikes typically precede loss spikes (but not every update spike triggers a loss spike); gradient norms often spike after update spikes; update spikes are larger in lower layers; and during update spikes the Adam first-moment estimate grows by roughly 2x while the second moment remains mostly unchanged, suggesting momentum buildup from unusually aligned gradients rather than a single outlier batch [26]. Issue #1390 documents an unrecovered spike after update clipping was inadvertently turned off, underscoring the brittleness of heuristic stabilizers at this scale [27]. Overall, these interventions reduced spike severity but did not remove the pathology, motivating the architectural QK-norm switch in Phase 3 [24].

4.4.2 Phase 2: Recovery attempts (discarded)

The team attempted short diagnostic restarts ("necromancy") and an optimizer swap (Muon). Muon temporarily reduced loss but eventually diverged, reinforcing the hypothesis that instability was rooted in the attention stack rather than optimizer state [24].

4.4.3 Phase 3: QK-norm switch

At step 80k, Marin switched to a Qwen3-style 32B configuration that adds QK-norm in attention and warm-started from the Llama 32B checkpoint. The switch imposed a one-time loss penalty but recovered within ~10B tokens; importantly, loss spikes disappeared entirely [8, 24].

4.4.4 Phase 4: Cooldowns (Bison vs Mantis) and shuffling

With stability restored, Marin ran cooldowns following the 8B playbook. The first cooldown (Bison) surfaced two issues: a GSM8K contamination incident and a shuffling-induced "phase shift" in training loss. The second cooldown (Mantis) fixed both by switching to a Feistel permutation (Section 3.6) and by cleaning the math component of the cooldown mix [24].

Attempt 1: Bison cooldown. Starting from the step-160k QK-norm checkpoint, Marin ran a 32k-step linear cooldown (160k → 192k; ~1.074T tokens) with a ~70/30 Nemotron/HQ mixture patterned after the 8B Starling cooldown (Table 8) [24]. The optimizer schedule used no warmup and a linear decay over the 160k→192k window, with AdamC enabled during decay and a small z-loss throughout [24]. Within the HQ budget, FLAN was upsampled 10x and an "all_math" Dolmino bundle 2x, mirroring the 8B recipe [24].

Dataset	Share
nemotron_cc/medium	21.49%
nemotron_cc/hq_synth	17.29%
nemotron_cc/medium_low	9.79%
nemotron_cc/hq_actual	5.81%
nemotron_cc/medium_high	5.24%
nemotron_cc/low_actual	4.46%
nemotron_cc/low_synth	3.99%
arxiv_markdownified	7.41%
dolmino/pes2o	7.41%
finemath-3-plus	4.33%
dolmino/flan	4.33%
stackexchange_custom	2.18%
dolmino/stackexchange	2.18%
starcoderdata	1.59%
all_math	1.08%
proofpile_2	0.35%
wikipedia_markdown	0.47%
dolmino/wiki	0.47%
medu_science_qa	0.15%

Table 8: Marin 32B Bison cooldown mixture (normalized share), reproduced from the retrospective [24].

Contamination: GSM8K. The retrospective traces the GSM8K anomaly to cached data: a Dolmino math bundle included GSM8K test items in a `test.json`, and although preprocessing was later updated to drop `test.json`, the old cached dataset persisted on the cluster and contaminated the Bison cooldown [24]. Because the contaminated GSM8K used OLMes formatting (not the LM Eval Harness default), contamination produced *worse* scores under the default prompt: the model assigned high surprisal to structured tags (e.g., `16-8=«16-7=9»9`) that were absent in the contaminated formatting, yielding extreme prompt fragility [24].

Shuffling: phase shift under linear permutation. Near step ~190k, the training loss jumped to a new plateau and never recovered, while evaluation losses remained flat (Figure 10) [24]. The retrospective interprets this as a data-ordering "phase shift" rather than instability: the underlying mixture weights did not change, but the affine/LCG permutation (Equation 1) can yield correlated stretches if the dataset blocks are structured and the stride is unlucky. The team notes that they originally chose the linear permutation because it is cheap and stateless (random access without permutation tables), and because per-component mixture sampling in Levanter keeps dataset proportions stable; in this case, those safeguards were insufficient [24].

Attempt 2: Mantis cooldown. Mantis restarted from step 160k with two targeted changes: (i) switching the sampling permutation to Feistel (Section 3.6), and (ii) replacing the Dolmino math bundle with MegaMath splits and later adding Common Pile EDU-filtered Python around step ~174k (renormalizing the HQ budget accordingly) [16, 24, 42]. With the optimizer schedule unchanged, both failure modes disappeared; empirically, the phase shift was removed and Paloma losses improved across corpora (Figure 10) [24].

Dataset	Share
nemotron_cc/medium	21.49%
nemotron_cc/hq_synth	17.29%
nemotron_cc/medium_low	9.79%
nemotron_cc/hq_actual	5.81%
nemotron_cc/medium_high	5.24%
nemotron_cc/low_actual	4.46%
nemotron_cc/low_synth	3.99%
megamath/web	5.57%
arxiv_markdownified	4.54%
megamath/text_code_block	4.24%
dolmino/pes2o	4.54%
megamath/web_pro	1.27%
megamath/translated_code	0.61%
megamath/qa	0.59%
finemath-3-plus	2.66%
dolmino/flan	2.66%
stackexchange_custom	1.34%
dolmino/stackexchange	1.34%
starcoderdata	1.59%
proofpile_2	0.35%
wikipedia_markdown	0.29%
dolmino/wiki	0.29%
medu_science_qa	0.09%

Table 9: Marin 32B Mantis cooldown mixture (normalized share), reproduced from the retrospective [24].

4.4.5 Base-model benchmark results (32B)

Table 10 reproduces a subset of the 32B retrospective results. Mantis improves substantially over Bison on math and code benchmarks and surpasses OLMo 2 32B base on average accuracy (with the caveats discussed above) [24, 29].

4.5 Open Development and Experiment History

A distinctive aspect of Marin is its public, issue-driven experimentation. The documentation includes an auto-generated summary of GitHub issues grouped by theme (quality classifiers, preprocessing, pretraining setups, SFT and domain-specific training) and a timeline of closed/open issues. Because that page is explicitly labeled as LLM-generated and "should not be trusted without verification," we treat it as a pointer to the underlying issues rather than a primary source [28].

Nevertheless, several themes recur across issues and in the 8B/32B retrospectives: (1) data extraction and formatting (HTML-to-text, Markdownification), (2) quality filtering signals (classifier-based and compression-based), (3) training-schedule tuning (WSD/WSD-S details), and (4) post-training tradeoffs (improving instruction scores without erasing base capabilities).

Model	Avg	MMLU	BBH	GPQA	MMLU-Pro	HumanEval	GSM8K	MATH
Marin 32B (Bison)	63.0	72.9	55.2	32.1	41.9	29.3	54.7	10.4
Marin 32B (Mantis)	65.2	74.7	59.6	34.0	45.1	42.7	69.1	15.3
OLMo 2 32B Base	63.2	71.9	56.1	32.2	42.0	23.8	76.4	12.7
Qwen 2.5 32B Base	68.1	80.8	67.4	39.0	57.9	48.8	89.3	36.3

Table 10: Selected 32B base-model results from the Marin 32B retrospective (LM Eval Harness defaults) [24].

5 Limitations and Conclusion

5.1 Limitations

Marin makes explicit several limitations common to fully open base-model releases.

- **Base-first focus.** Marin 32B is released only as a base model without instruction tuning or RLHF, limiting immediate end-user utility [24].
- **Evaluation uncertainty.** Many benchmarks are contaminated in modern pretraining corpora; comparisons should be interpreted cautiously and ideally validated with decontaminated or held-out alternatives [24, 25].
- **Language and context scope.** The reported results focus on English (and code) and do not include long-context extension training.
- **Operational drift.** Mid-flight adaptation is powerful, but it also means that the final recipe is the result of multiple reactive decisions; reproducing the trajectory requires careful artifact/version tracking.

5.2 Conclusion

Marin is a fully open effort to train strong base language models while keeping the training process legible: data mixes, code, checkpoints, evaluation settings, and a public record of issues and fixes [13]. Across an 8B run that evolves through multiple cooldown/reheat phases and a 32B run that required a mid-training attention-stack change, the retrospectives highlight several transferable lessons:

- **Mid-flight changes can work.** Carefully staged transitions (cooldowns, reheats, and mix shifts) can be executed without catastrophic regressions when instrumentation is strong.
- **Architectural headroom matters at scale.** At 32B, optimizer-side heuristics reduced but did not eliminate loss spikes; QK-norm provided the needed stability margin after a short recovery window [8, 24].
- **Deep cooldowns surface new failure modes.** Very low learning rates exposed `lm_head` instability in 8B cooldowns; z-loss was an effective practical fix [4, 6, 25, 40].
- **Shuffling is not a solved detail.** A valid but poorly mixing permutation can produce late-training pathologies; a Feistel-based shuffle resolved a measurable phase shift in 32B cooldown [24].

Future work. The retrospectives suggest several next steps: improving the 8B/32B post-training pipeline (SFT with base-retention, preference tuning, RL), adding long-context extension stages, and hardening dataset auditing (especially for contamination and caching hazards).

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[2] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[3] Cody Blakeney, Mansheej Paul, Brett W. Larsen, Sean Owen, and Jonathan Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training, 2024. URL https://arxiv.org/abs/2406.03476.

[4] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *ArXiv*, abs/2405.09818, 2024. URL https://api.semanticscholar.org/CorpusID:269791516.

[5] 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.

[6] 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 M. 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 García, 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 Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311, 2022. URL https://api.semanticscholar.org/CorpusID:247951931.

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

[8] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim M. Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Paveti'c, Dustin Tran, Thomas Kipf, Mario Luvci'c, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. *ArXiv*, abs/2302.05442, 2023. URL https://api.semanticscholar.org/CorpusID:256808367.

[9] Gemma 3 Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

[10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathurx, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov,

Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao

Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[11] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, A. Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Daniel Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke S. Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hanna Hajishirzi. Olmo: Accelerating the science of language models. *ArXiv*, abs/2402.00838, 2024. URL https://api.semanticscholar.org/CorpusID:267365485.

[12] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

[13] David Hall, Christopher Chou, Abhinav Garg, Nikil Ravi, Nelson Liu, Herumb Shandilya, Ahmed Ahmed, Percy Liang, Rohith Kuditipudi, J38, Tony Lee, Russell Power, Kamyar Salahi, William Held, Jason Wang, chiheem, Joel Niklaus, Yifan Mai, dependabot[bot], Ivan Zhou, Kevin Xiang Li, Sherry Yang, Sidd Karamcheti, Ryan Williams, Cathy Zhou, Ashwin Ramaswami, whenwen, Suhas Kotha, Gary Miguel, and Calvin Xu. marin-community/marin. https://github.com/marin-community/marin, nov 14 2025. URL https://github.com/marin-community/marin.

[14] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[15] 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. *NeurIPS*, 2021.

[16] Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A. Feder Cooper, Aviya Skowron, John Kirchenbauer, Shayne Longpre, Lintang Sutawika, Alon Albalak, Zhenlin Xu, Guilherme Penedo, Loubna Ben Allal, Elie Bakouch, John David Pressman, Honglu Fan, Dashiell Stander, Guangyu Song,

Aaron Gokaslan, Tom Goldstein, Brian R. Bartoldson, Bhavya Kailkhura, and Tyler Murray. The common pile v0.1: An 8tb dataset of public domain and openly licensed text, 2025. URL https://arxiv.org/abs/2506.05209.

[17] Nathan Lambert, Jacob Daniel Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hanna Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. 2024. URL https://api.semanticscholar.org/CorpusID:274192505.

[18] Levanter Authors. Levanter data shuffling via pseudo-random permutations (linear and feistel). Source code (GitHub), 2025. URL https://github.com/stanford-crfm/levanter/blob/9fde0781a1737e088535c392cf239aba5e1143e2/src/levanter/data/_prp.py. Commit 9fde0781; accessed: 2026-02-05.

[19] 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.

[20] Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023.

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

[22] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*, 2024.

[23] Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hannaneh Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. Paloma: A benchmark for evaluating language model fit, 2024. URL https://arxiv.org/abs/2312.10523.

[24] Marin Community. Marin 32b retro. Marin Documentation, 2025. URL https://marin.readthedocs.io/en/latest/reports/marin-32b-retro/. Accessed: 2026-02-05.

[25] Marin Community. Marin 8b retro. Marin Documentation, 2025. URL https://marin.readthedocs.io/en/latest/reports/marin-8b-retro/. Accessed: 2026-02-05.

[26] Marin Community. Experiment: Debug 32b spiking (github issue 1368). GitHub issue, 2025. URL https://github.com/marin-community/marin/issues/1368. Accessed: 2026-02-05.

[27] Marin Community. Experiment: Revive the 32b (github issue 1390). GitHub issue, 2025. URL https://github.com/marin-community/marin/issues/1390. Accessed: 2026-02-05.

[28] Marin Community. Auto-generated summary (github issues report). Marin Documentation, 2025. URL https://marin.readthedocs.io/en/latest/reports/summary/. Accessed: 2026-02-05.

[29] 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, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, 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.

[30] OpenAI. Introducing improvements to the fine-tuning API and expanding our custom models program, 4 2024. URL https://openai.com/index/introducing-improvements-to-the-fine-tuning-api-and-expanding-our-custom-models-program/.

[31] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

[32] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

[33] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024.

[34] Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset, 2025. URL https://arxiv.org/abs/2412.02595.

[35] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[36] Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models, 2024. URL https://arxiv.org/abs/2312.16903.

[37] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

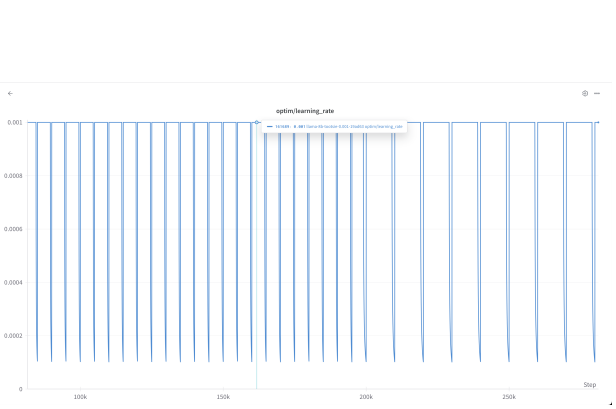
[38] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

[39] Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape perspective, 2024. URL https://arxiv.org/abs/2410.05192.

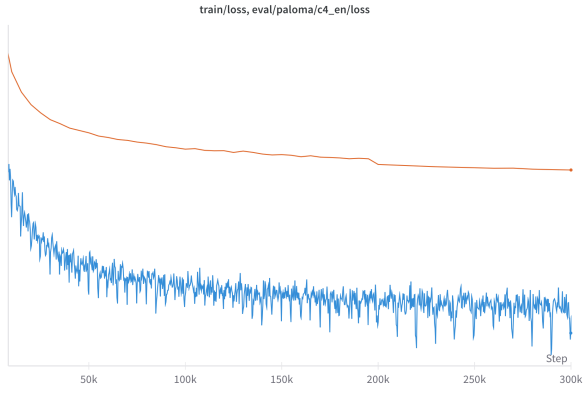
[40] Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities, 2023. URL https://arxiv.org/abs/2309.14322.

[41] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

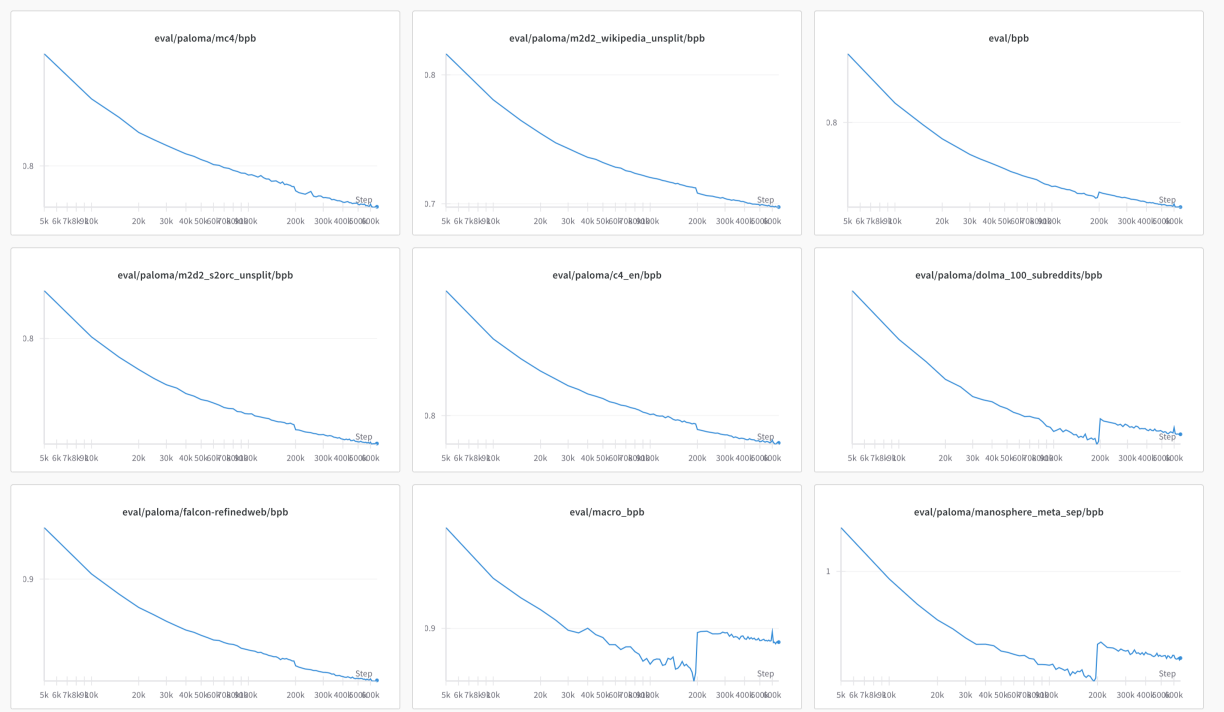
[42] Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P. Xing. Megamath: Pushing the limits of open math corpora, 2025. URL https://arxiv.org/abs/2504.02807.



(a) Decay-cycle spacing change.



(b) Eval/training loss drops during decay.



(c) Diverse eval-loss trajectories post-transition.

Figure 2: WSD-S diagnostics from Marin 8B Phase 1 (reproduced) [25].

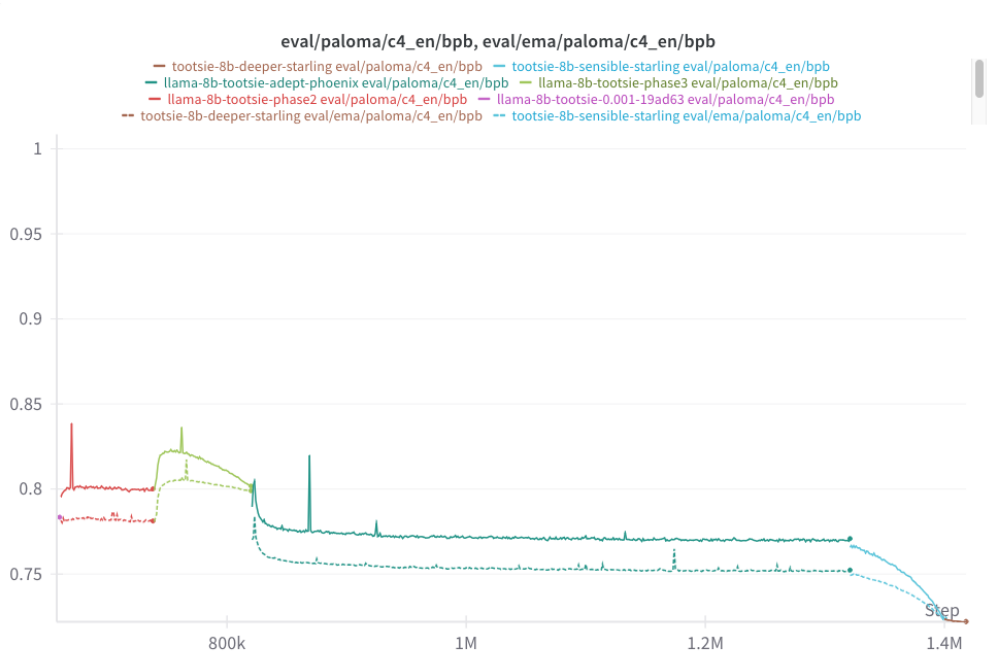


Figure 3: The "EMA gap" during Ocelot: the difference between hot-model and EMA-model evaluation loss remains surprisingly stable at high learning rates (reproduced) [25].

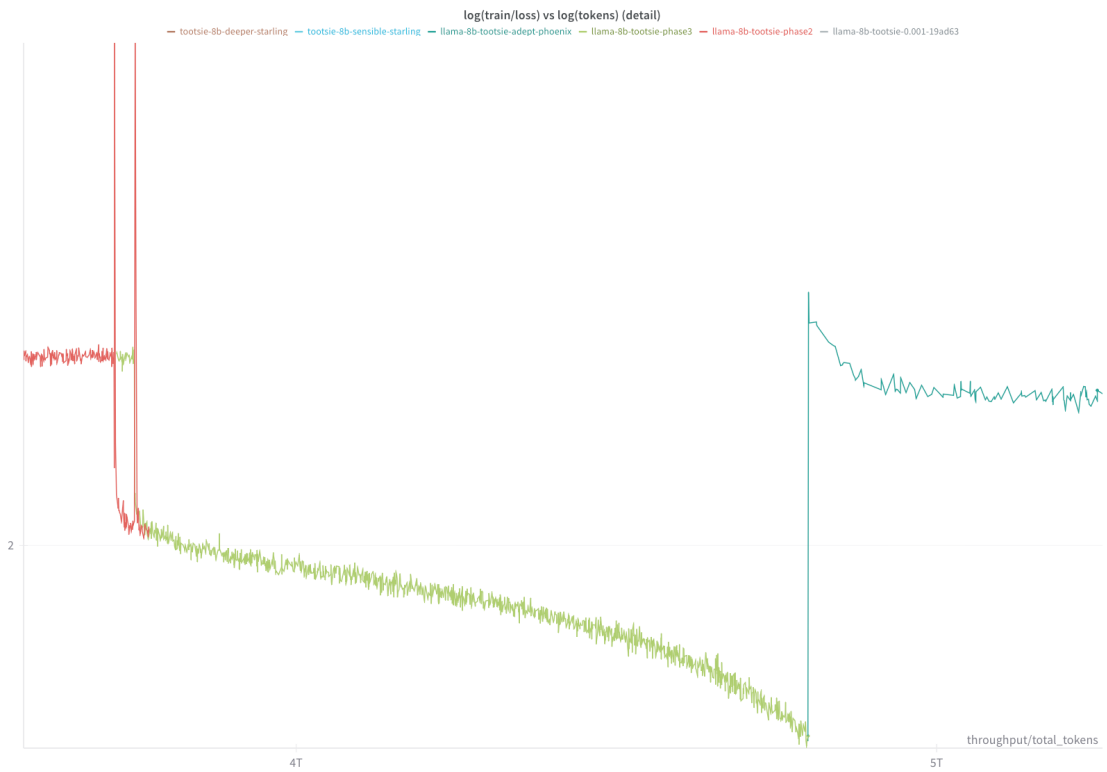
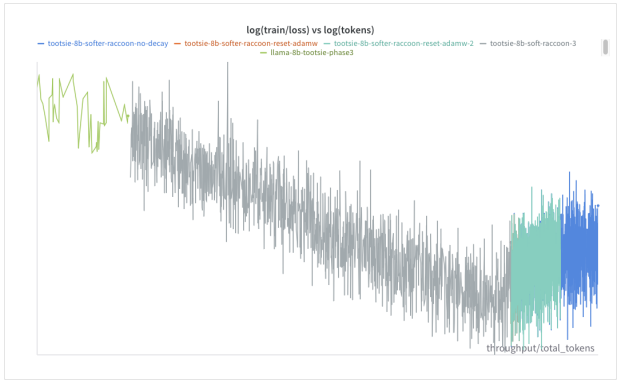
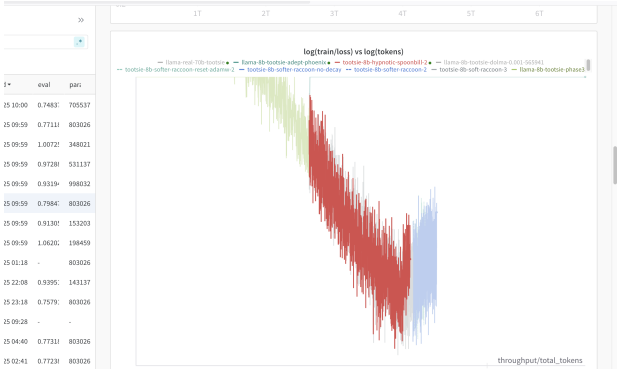


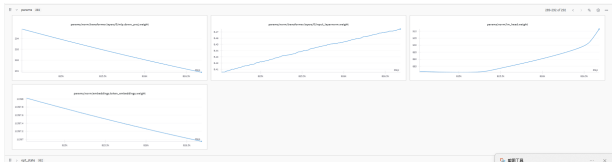
Figure 4: Phoenix transition loss curve: a brief spike followed by recovery to slightly better loss than pre-cooldown (reproduced) [25].



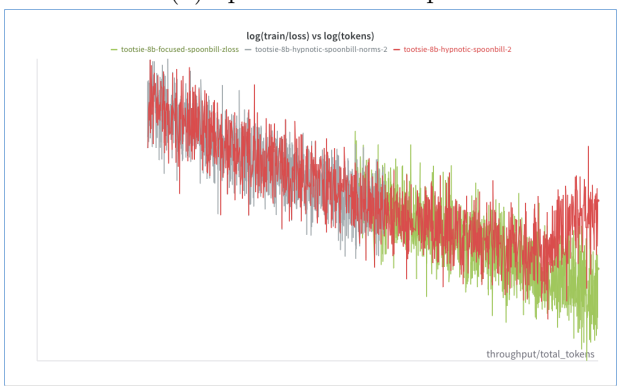
(a) Raccoon loss creep.



(b) Spoonbill loss creep.

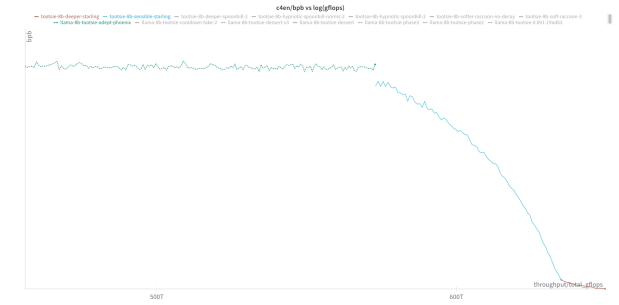


(c) `lm_head` norm explosion.

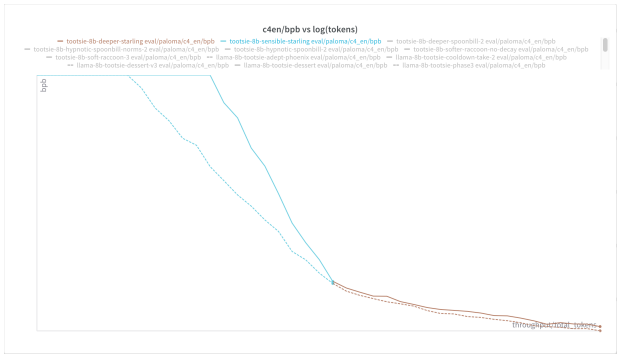


(d) z-loss fix.

Figure 5: Deep cooldown stability: loss creep and the z-loss intervention (reproduced) [25].

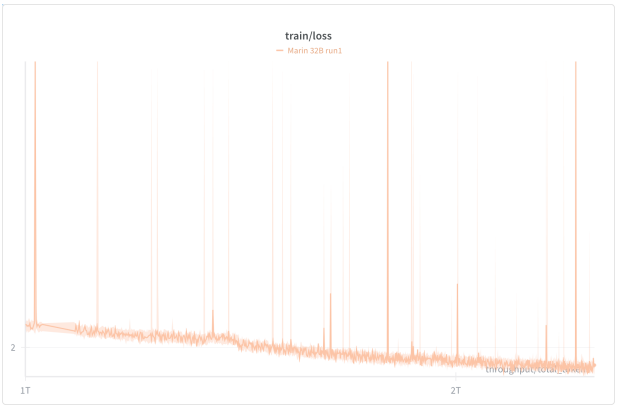


(a) `c4_en` perplexity.

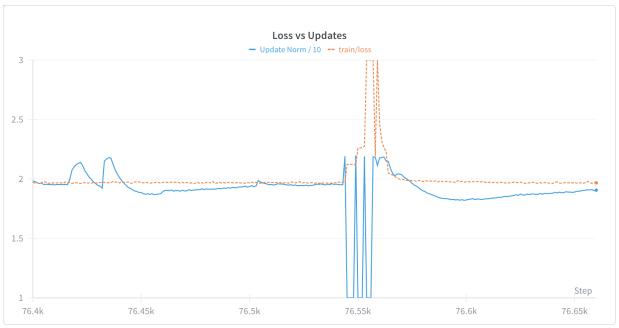


(b) Loss slowdown at fixed low LR.

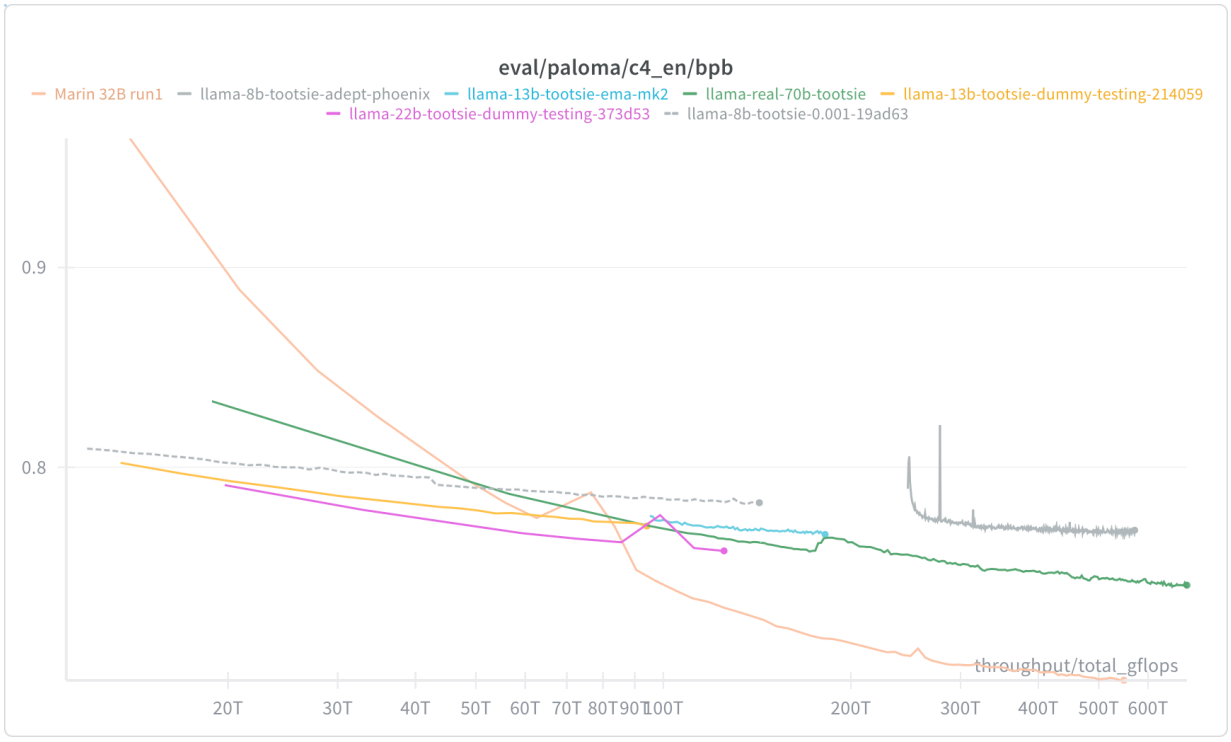
Figure 6: Starling cooldown diagnostics (reproduced) [25].



(a) Training loss spikes.



(b) Update spikes precede loss spikes.



(c) Eval-loss comparisons.

Figure 7: Marin 32B Phase 1 instability diagnostics (reproduced) [24].

(a) Restart spike. (b) Muon vs Adam. (c) Muon divergence.

Figure 8: Phase 2 recovery attempts (reproduced) [24].

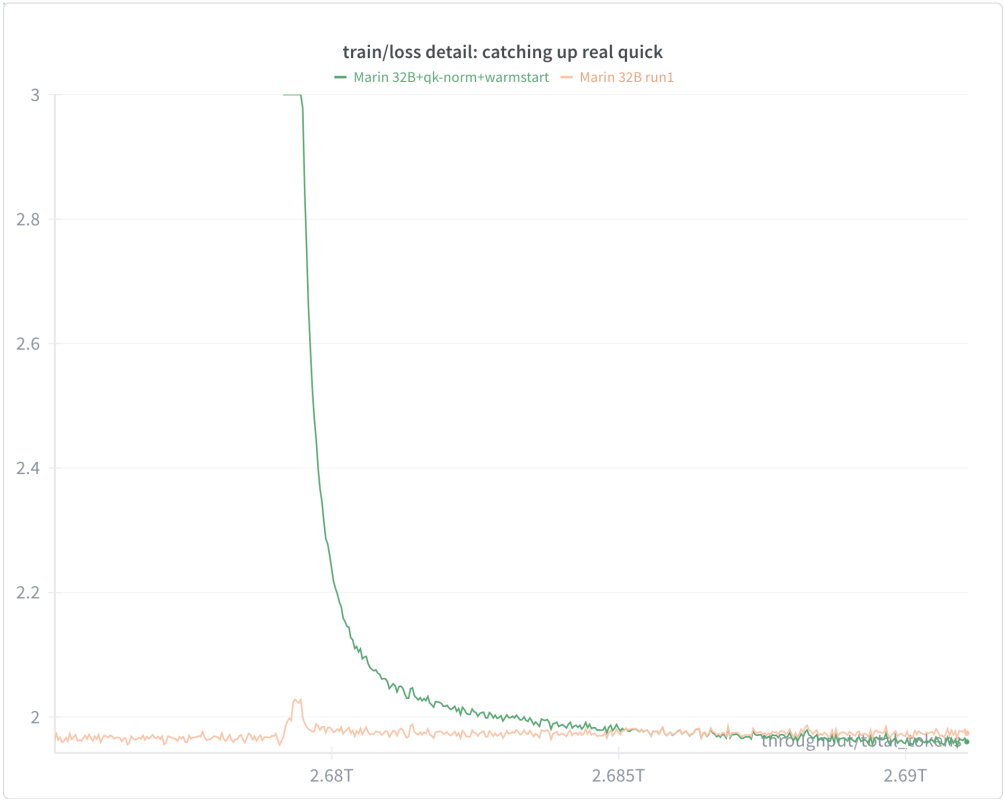
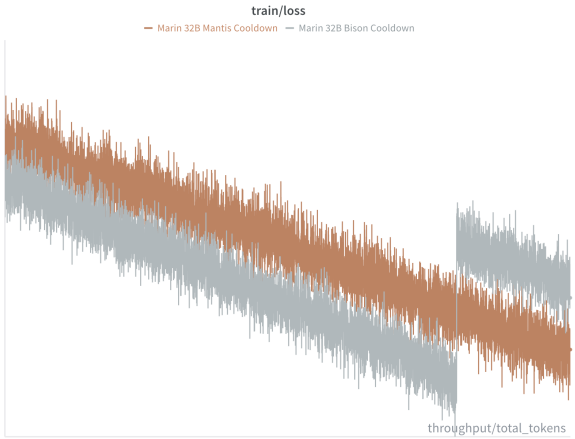
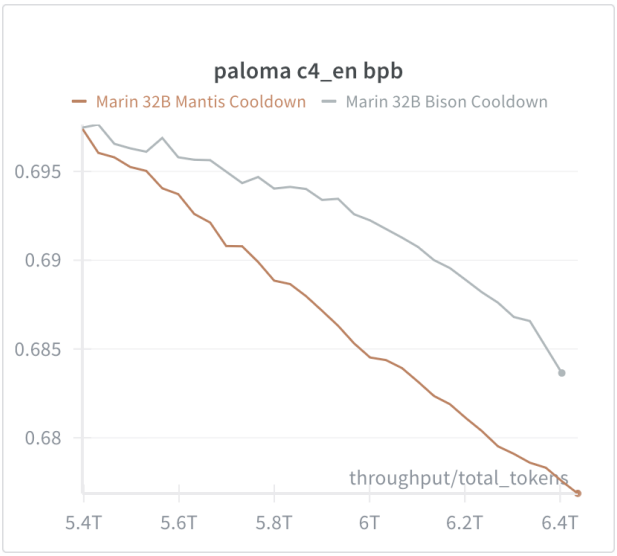


Figure 9: QK-norm warm-start recovery: loss returns to the pre-switch trajectory after a short recovery window (reproduced) [24].

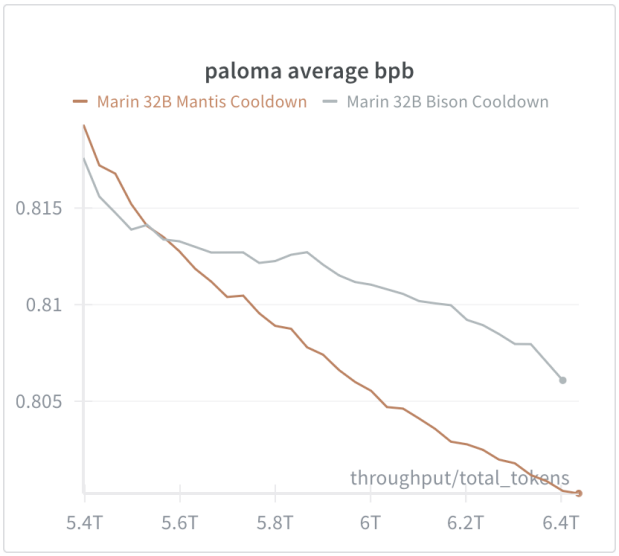
(a) Cooldown phase shift (LCG).



(b) Feistel removes shift.



(c) Paloma `c4_en`.



(d) Paloma average.

Figure 10: Shuffling pathology and fix during Marin 32B cooldowns (reproduced) [24].