MEGREZ: RECONSTRUCTING INTERMEDIATE SUPERVISION FOR TERNARY LOOPED TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ternary looped transformers reuse the same quantized weights across recurrent calls, coupling the approximation of intermediate teacher states. We present Megrez, a calibration framework that adapts the supervision targets to the current student. For each token, Megrez jointly transforms normalized teacher states with a single orthogonal map, minimizing alignment error within a family that preserves the teacher's output under its unchanged native readout. The constrained alignment has an exact low-dimensional solution. Megrez alternates target reconstruction with shared ternary weight updates, accommodating gated and terminal readouts without changing the inference architecture. Across Ouro-1.4B, Parcae-1.3B and LoopFormer, reconstruction reduces squared target error by 20.2–55.5% on average per window at fixed student weights. On Ouro-1.4B, the complete framework lowers C4 perplexity from 45.44 to 29.54 and raises HellaSwag accuracy from 39.83% to 44.92% compared with the restricted TernaryLLM adaptation.

1 INTRODUCTION

Looped transformers increase computational depth by repeatedly applying the same parameterized block, as in Ouro (Zhu et al., 2026), Parcae (Prairie et al., 2026) and LoopFormer (Jeddi et al., 2026). Ternary quantization further reduces weight storage. Calibrating the resulting model requires a shared set of three-level weights to process changing hidden states. Errors at one call can change later inputs, coupling the fit across recurrent calls.

Output matching trains a student to reproduce teacher predictions (Hinton et al., 2015), while intermediate matching adds teacher hidden states as supervision (Romero et al., 2015; Sun et al., 2019). With fixed intermediate targets, one set of ternary weights must approximate the teacher's representations at every recurrent call. Our goal is to adapt these targets to the current student while preserving the teacher's output under its unchanged *native readout*, the original computation from recurrent states to logits.

Consider a teacher that reads the final recurrent state. Holding this state fixed preserves the readout on a constructed target sequence while leaving earlier targets available for alignment. Ouro combines recurrent states using gates computed from those states. Preserving an old weighted sum can therefore fail if target changes alter the gates. Fixing both the gate scores and the weighted aggregate preserves its native output.

We present Megrez, a calibration framework that reconstructs supervision targets for the current student. For each token, one orthogonal map jointly transforms its normalized teacher states across recurrent calls, preserving their pairwise inner products and fixing the required readout directions. We solve the constrained alignment exactly in a small subspace using orthogonal Procrustes algebra (Schönemann, 1966). The original targets are feasible, so the exact solution cannot increase alignment error at a fixed student snapshot. Megrez alternates target reconstruction with shared ternary weight updates and output matching (Figure 1). The targets are refreshed during calibration, and the transformations are discarded at inference.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

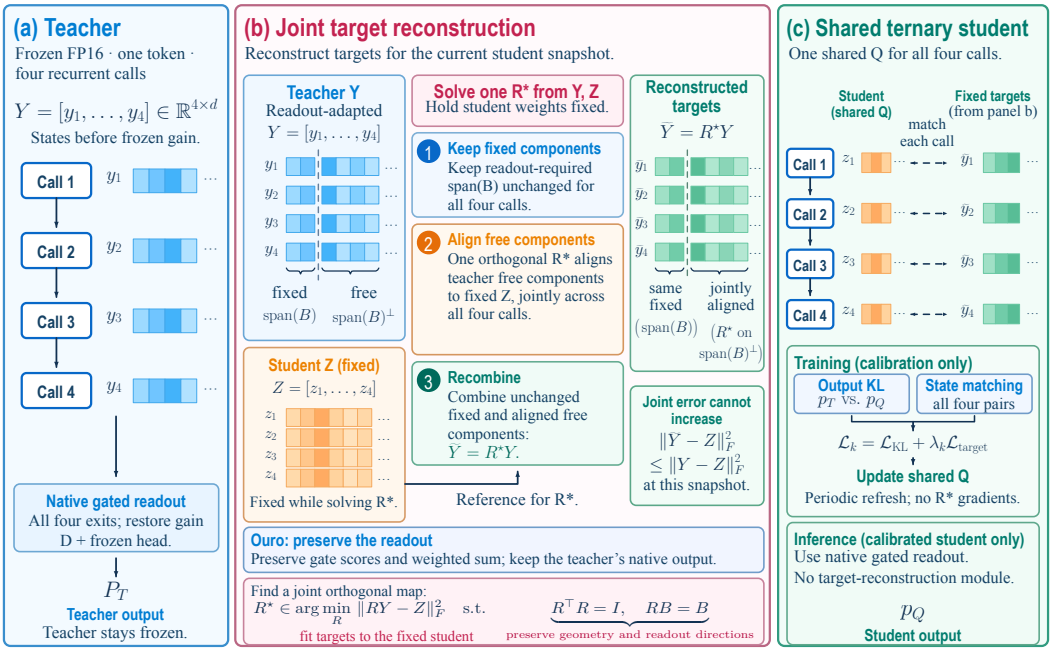

Figure 1: Megrez on Ouro for one token ($K = 4$). (a) The frozen teacher supplies recurrent states. (b) Joint orthogonal reconstruction preserves the native readout without increasing pre-gain target error at a fixed student snapshot. (c) Shared ternary calibration combines output KL with periodically refreshed, detached state targets. Inference retains the native student readout.

We evaluate Megrez on Ouro-1.4B, Parcae-1.3B and LoopFormer 3×8. At fixed student weights, reconstruction reduces squared target error by 20.2–55.5% on average per window. Matched controls separately assess the predictive effect of reconstructed-target supervision (Table 2). Target-family ablations show different rankings for alignment, perplexity and accuracy (Section 4.4).

The complete framework improves all three perplexities and six of seven task accuracies over restricted TernaryLLM (Chen et al., 2024) on Ouro-1.4B, including a reduction in C4 perplexity from 45.44 to 29.54. Parcae-1.3B and LoopFormer improve WikiText-2 (WT2) (Merity et al., 2016) and C4 perplexity over their TWLA-T (Zhao et al., 2026) initializers.

Our contributions are:

- **Adaptive supervision for ternary recurrence.** We introduce Megrez, which alternates joint target reconstruction with shared ternary weight updates while preserving the teacher's native output on the reconstructed targets.
- **Exact alignment with readout constraints.** For gated and final-state readouts, we derive an exact low-dimensional solution that preserves normalized teacher geometry and cannot increase target error at fixed student weights.
- **Evaluation across recurrent architectures.** Reconstruction reduces squared target error by 20.2–55.5% on average per window across three architectures. Matched controls and target-family ablations distinguish alignment gains from predictive effects.

2 Related Work

Recurrent computation and quantization. Universal Transformers established recurrence over depth (Dehghani et al., 2019); Ouro, Huginn, Parcae and LoopFormer develop shared-parameter latent computation (Zhu et al., 2026; Geiping et al., 2025; Prairie et al., 2026; Jeddi et al., 2026). LoopQ combines loop-dependent scaling, transforms, adapters and trajectory distillation with adaptive loss weighting (Fang et al., 2026). Megrez instead reconstructs the intermediate targets under native-readout constraints while calibrating shared ternary weights.

Ternary representations and calibration. BitNet b1.58 trains ternary models from scratch (Ma et al., 2024); TernaryLLM, PT^2-LLM and CAT-Q calibrate pretrained weights through learned scales/shifts, asymmetric grids and softened ternarization (Chen et al., 2024; Yan et al., 2026; Wang et al., 2026b). TWLA also quantizes activations (Zhao et al., 2026). Megrez and the matched KL-only controls use PV-Tuning's discrete and continuous weight updates (Malinovskii et al., 2024). Megrez additionally reconstructs the supervision targets.

Rotations for compression. QuIP, QuaRot and SpinQuant change quantization coordinates through corresponding weight and activation transformations (Chee et al., 2023; Ashkboos et al., 2024; Liu et al., 2025). Megrez transforms supervision targets with the teacher readout fixed; these maps disappear after calibration. Retained initializer transforms have a separate deployment role.

Representation alignment and adaptive supervision. VkD learns an orthogonal student-side projector; geometry-aware distillation uses Gram and Procrustes objectives (Miles et al., 2024; Bhattarai et al., 2025). SimKD reuses a teacher classifier on aligned student features (Chen et al., 2022); AID fine-tunes the teacher (Qian et al., 2025). Representation-equivalence analysis jointly reparameterizes representations and their reader (Han, 2026). Megrez jointly transforms a token's recurrent targets under a fixed native readout, preserving gate/aggregate directions or the final state. Classical Procrustes algebra supplies the solve (Schönemann, 1966); the readout defines its admissible target family. Appendix J distinguishes this within-token geometry from conventional and relational distillation.

3 MEGREZ

Megrez first constructs targets for the current student, then updates the shared ternary weights using those targets. The targets preserve the teacher's output under its original readout. Ouro requires both its gates and weighted sum of states to remain fixed; Appendix E shows why preserving the old weighted sum alone is insufficient.

3.1 AN ADMISSIBLE TARGET FAMILY

Consider a sampled token position i. After recurrent call t, the teacher and student have hidden states $h^T_{t,i}$ and $h^Q_{t,i}$. For example, Ouro produces four teacher states and four student states for this same position ($K = 4$). After the normalization defined below, we collect them as columns of $Y_i = [y_{1,i}, \ldots, y_{K,i}]$ and $Z_i = [z_{1,i}, \ldots, z_{K,i}] \in \mathbb{R}^{d\times K}$, where d is the hidden dimension. Each column corresponds to a recurrent call. We omit i while solving for one token.

A single $d \times d$ orthogonal map R transforms all K teacher columns together. Such a map preserves their lengths and pairwise inner products in normalized coordinates, preventing a trivial reduction in distance by shrinking the targets. Let $B \in \mathbb{R}^{d\times q}$ have orthonormal columns spanning the readout directions to be fixed. We reconstruct targets by solving

$$R^\star \in \arg\min_{R\in\mathcal{G}_B} \|RY - Z\|_F^2, \qquad \mathcal{G}_B = \{R : R^\top R = I,\ RB = B\}. \tag{1}$$

The constraint $RB = B$ leaves every direction in B unchanged; the readout determines these directions. Each token has its own $R^\star$. The resulting family is sufficient to preserve the readout, but may exclude other output-preserving transformations.

The alignment error splits into a part along the fixed directions and a part in the remaining directions. With $P = I - BB^\top$, orthogonality and $RB = B$ give

$$\|RY - Z\|_F^2 = \|B^\top(Y - Z)\|_F^2 + \|RPY - PZ\|_F^2. \tag{2}$$

The first term is teacher–student mismatch that no allowed R can change. The second is the mismatch that target reconstruction can reduce.

Figure 2 makes this freedom visible for one token with three states. Fixing its final state leaves a common transformation in the other two dimensions: intermediate targets can move toward the student while the final-state readout stays unchanged. The error curves show how choosing that transformation improves alignment.

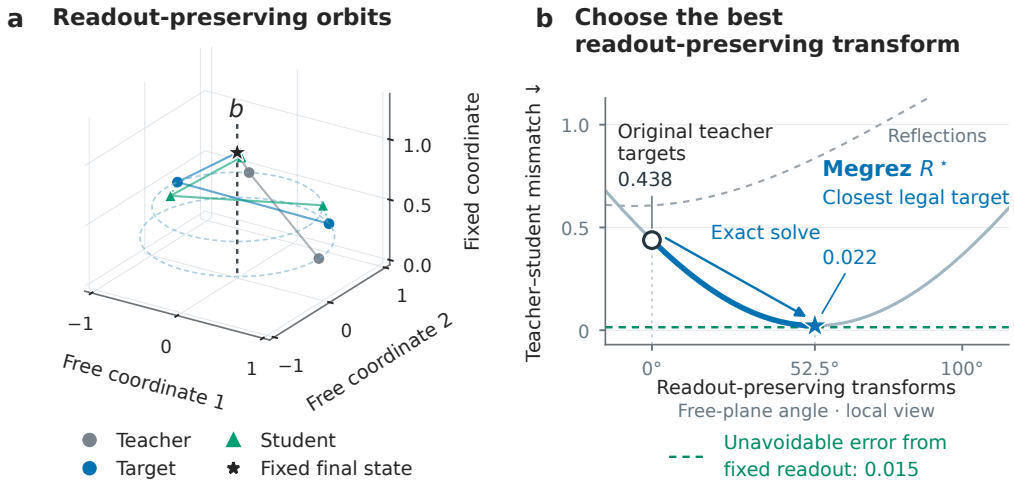

Figure 2: How targets can move while the readout stays fixed. Analytical example for one token ($d = 3$, $K = 3$). **(a)** One orthogonal map moves the intermediate teacher states along dashed orbits, fixing $b = y_K = (0, 0, 1)^\top$ and preserving pairwise inner products. **(b)** Local view of readout-preserving transforms. The exact solve selects $R^\star$ from both full rotation and reflection families, reducing normalized squared mismatch from 0.438 to 0.022. The green dashed line marks the unavoidable fixed-readout component (0.015).

3.2 Constraints from the native readout

Ouro: gated multi-exit readout. Let h_t^T, h_t^Q be teacher and student exit states after final RMSNorm (Zhang & Sennrich, 2019) and its learned per-coordinate gain, with $K = 4$. Assume the frozen $D = \mathrm{diag}(\gamma)$ is invertible and set $y_t = D^{-1}h_t^T$, $z_t = D^{-1}h_t^Q$. Removing the diagonal gain defines the coordinates in which the transformation is orthogonal. The frozen gate vector v, bias β and sigmoid σ define

$$a_t = \sigma(v^\top h_t^T + \beta), \quad \alpha_t = a_t \prod_{j<t}(1 - a_j)\ (t < K), \quad \alpha_K = \prod_{j<K}(1 - a_j). \tag{3}$$

Here α_t is the weight of exit t; the last exit receives the remaining weight. With fixed head W_o, native aggregate logits are $\ell^T = W_o \sum_t \alpha_t h_t^T$. Define $\mu = \sum_t \alpha_t y_t$ and $\nu = Dv$, and choose $\mathrm{span}(B) = \mathrm{span}\{\mu, \nu\}$. Fixing ν keeps the gates unchanged; fixing μ then keeps their weighted aggregate unchanged (see the aggregate-only counterexample in Appendix E). Raw targets are $\widetilde{h}_t^T = DR^\star y_t$. These constraints preserve the aggregate without requiring a nontrivial nullspace of W_o, while allowing individual exit targets to change.

Parcae and LoopFormer: final-state readout. Here h_t^T, h_t^Q are captured after each of eight recurrent core or stack applications, before the downstream readout. Define $\rho(h) = \sqrt{\|h\|_2^2/d}$ for nonzero states, and use $y_t = h_t^T/\rho(h_t^T)$, $z_t = h_t^Q/\rho(h_t^Q)$. Choose $\mathrm{span}(B) = \mathrm{span}\{y_K\}$ and restore teacher RMS: $\widetilde{h}_t^T = \rho(h_t^T)R^\star y_t$. Then $\widetilde{h}_K^T = h_K^T$. Fixing this state preserves the teacher's readout through Parcae's post-recurrence layers (coda) and output head, or LoopFormer's final normalization and head, when applied to the constructed target sequence. This also covers a coda that mixes tokens because the complete final-state sequence is unchanged. The other seven states remain available for alignment. Numerical RMS floors and final-anchor correction are specified in Appendix B.

3.3 Properties and reduced solve

Proposition 1 (target preservation). In exact arithmetic, every feasible R preserves $Y^\top Y$. The Ouro instance also preserves gate scores, exit weights and aggregate logits; the final-state instance

preserves the fixed downstream readout. For Ouro, orthogonality and $R\nu = \nu$ preserve gate scores and exit weights. Fixing μ then gives

$$\sum_t \alpha_t \widetilde{h}_t^T = DR \sum_t \alpha_t y_t = D\mu = \sum_t \alpha_t h_t^T. \tag{4}$$

A fixed head bias is preserved as well. The inner products are preserved in normalized coordinates; Ouro's per-coordinate gain generally prevents the same guarantee for raw exit states.

Proposition 2 (alignment and computation). An optimum exists and satisfies $\|R^\star Y - Z\|_F^2 \le \|Y - Z\|_F^2$, since the original targets ($R = I$) are feasible. For the solve, let U be an orthonormal basis of $\mathrm{span}[PY, PZ]$, with dimension $r \le 2K$. The teacher and student coordinates in this basis are $X = U^\top PY$ and $T = U^\top PZ$, respectively. Compute

$$TX^\top = L\Sigma V^\top, \qquad O = LV^\top, \qquad R^\star = I + U(O - I_r)U^\top. \tag{5}$$

The Procrustes solution (Schönemann, 1966) allows reflections and acts as the identity outside the active subspace. It costs $O(dK^2 + K^3)$ per sampled token and requires no $d \times d$ matrix. Appendix E proves the solve for all ranks, gives the exact attainable gain, and characterizes when the original targets are optimal.

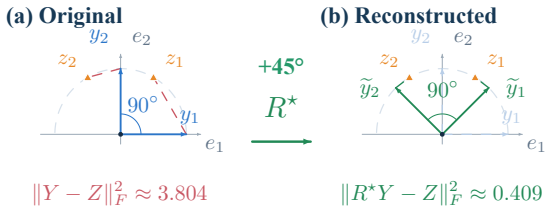

Figure 3: Joint reconstruction in normalized coordinates. A shared 45° rotation maps teacher states (blue) to targets (green), preserving norms, their 90° angle, and the final state along e_3. With student states (orange) fixed, squared fitting error falls from 3.804 to 0.409. Dashed segments show residuals.

Figure 3 illustrates the solution for $d = K = 3$ and $B = e_3$, with $y_K = \widetilde{y}_K = z_K = \sqrt{3}e_3$ fixed. Calibration then optimizes their joint fit through shared ternary weights. Section 4.3 measures the resulting predictive effects. If $K = 1$ and that sole state is fixed, no target movement is possible.

3.4 Calibration and Deployment

The student uses one quantized parameter set for all recurrent calls. For each group of 128 weights, its affine ternary representation is

$$Q_g = b_g \mathbf{1} + s_g c_g, \qquad c_g \in \{-1, 0, 1\}^{128}, \quad s_g > 0. \tag{6}$$

Here c_g contains the ternary codes, b_g is the offset and s_g is the scale; all are shared across calls. Ouro, Parcae and LoopFormer quantize 168, 144 and 12 physical projections, respectively; native unquantized components and fixed initializer transforms remain in the deployment payload (Appendix B).

At weight-update step k, let $\mathcal{P}$ be the sampled token positions in a calibration window (a fixed-length token sequence). The target $\widetilde{y}_{t,i} = R_i^\star y_{t,i}$ is held fixed, while the student's normalized state $z_{t,i}(Q)$ depends on its weights Q. We use

$$\mathcal{L}_k(Q) = \mathcal{L}_{\mathrm{KL}}(Q) + \lambda_k \frac{\sum_{i \in \mathcal{P}, t} \|z_{t,i}(Q) - \widetilde{y}_{t,i}\|_2^2}{\sum_{i \in \mathcal{P}, t} \|y_{t,i}\|_2^2}, \tag{7}$$

where $\mathcal{L}_{\mathrm{KL}}$ matches the student's output distribution to the teacher's using their native readouts, and λ_k weights the intermediate-state loss. Gradients update Q without passing through target reconstruction; this is what we mean by *detached targets*. Ouro divides both raw residuals and teacher states by the frozen gain. The final-state instances normalize each student state by its own RMS. Their implementations renormalize reconstructed targets before caching, which agrees with the formula in exact arithmetic.

Calibration procedure. Every eight updates, the primary recipes run the teacher and current hard-quantized student on the next eight calibration windows. For each sampled token, they form Y, Z, B,

solve Eq. (5), and cache the reconstructed targets. These targets remain fixed for the next eight weight updates in Eq. (7), after which reconstruction uses the updated student. Figure 1 summarizes the Ouro workflow; Appendix E gives the full algorithm. Each main Megrez endpoint uses 256 updates, a linearly decaying auxiliary coefficient from 0.03 to zero, and cosine-decayed weight-update rates. Token sampling and initialization differ by model; Appendix B gives the settings and the Ouro FP32-gradient/FP16-deployment boundary.

Weight optimization follows PV-Tuning's discrete-assignment and continuous- parameter updates (Malinovskii et al., 2024). The Parcae implementation uses Adam (Kingma & Ba, 2015) for log scales and LAMB (You et al., 2020) for the remaining parameters. The matched KL-only control sets $\lambda_k = 0$ while retaining the initializer, data, numerical policy and update budget, isolating the added supervision within the same calibration pipeline.

At inference, only the calibrated student remains, including its native unquantized components and retained initializer transforms. The teacher, target cache and target transformations are discarded. Appendix C.5 reports storage and forward runtime.

4 Experiments

We evaluate the complete recipe, target reconstruction on fixed student states, and predictive effects under matched training controls.

4.1 Experimental setup

We evaluate Ouro-1.4B (Zhu et al., 2026), Parcae-1.3B (Prairie et al., 2026) and LoopFormer 3×8 (Jeddi et al., 2026) with four/eight/eight recurrent calls. Megrez uses shared, group-128 ternary weights with 16-bit activations at deployment. LoopQ-T retains group size 32, CTA and per-loop copies, counted in storage.

Primary tables compare FP16/BF16, GPTQ-T (Frantar et al., 2023), TernaryLLM (Chen et al., 2024), CAT-Q-T (Wang et al., 2026b), PT^2-LLM-T (Yan et al., 2026), TWLA-T (Zhao et al., 2026), LoopQ-T (Fang et al., 2026) and Megrez; T denotes our ternary adaptation. Primary Megrez cohorts use 128 windows, 256 updates and seeds 20260909–20260911, starting from optimized native-PV on Ouro or single-plane TWLA-T on Parcae/LoopFormer, retaining initializer transforms. Recipe comparisons have unequal budgets and initialization; restricted Ouro TernaryLLM matches windows, updates and seeds, but not initialization. Appendices C.4 and D specify adaptations and costs.

We report perplexity on WT2, C4 (Raffel et al., 2020) and LAMBADA (Paperno et al., 2016). Accuracy covers HellaSwag (Zellers et al., 2019), ARC-Challenge/Easy (Clark et al., 2018), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), LAMBADA and MMLU (Hendrycks et al., 2021), using complete zero-shot splits except five-shot MMLU. Appendix A specifies checkpoints, task normalization and fixed evaluation populations. Means and sample SD describe training-seed variation on fixed evaluation sets. Endpoints use the final update without test-based seed selection. LoopFormer's initializer was chosen using test results (Appendix D). Tables disclose seed counts; dashes mark unavailable values.

4.2 Quality of the complete calibration recipe

Gated multi-exit calibration. On Ouro-1.4B, Megrez reduces WT2/C4 perplexity from its optimized native-PV parent's 29.37/33.44 to 17.84/29.54. The parent includes two trained projection overlays; its matched output-only continuation isolates the added supervision.

Against an external recipe (Table 1), Megrez reduces WT2/C4 perplexity from the restricted TernaryLLM adaptation's 20.94/45.44 to 17.84/29.54, reductions of 14.8% and 35.0%. It also improves LAMBADA perplexity and six of seven task accuracies, including HellaSwag (39.83% to 44.92%). Both methods use the same 818.431 MB active tensor payload.

TernaryLLM exceeds Megrez on WinoGrande, while LoopQ-T is higher on MMLU. The floating-point teacher scores WT2/C4 8.60/12.61. Other adapted recipes produce very large perplexities under the evaluated configurations. Table 7 specifies their representations and retained components.

Table 1: Complete-recipe results across three recurrent architectures, with supplementary Ouro-2.6B evaluations in Appendix I. Megrez and Ouro TernaryLLM average three seeds; other rows are single endpoints. **Bold** marks the best quantized point estimate per column. T denotes a ternary adaptation; see Table 7 and Appendix D for recipe details.

	Perplexity ↓			Accuracy (%) ↑						
Method	WT2	C4	LamP	HS	ARC-C	ARC-E	PIQA	WG	Lam	MMLU
Ouro-1.4B										
FP16/BF16	8.60	12.61	5.33	71.65	52.82	76.39	78.46	66.14	65.24	67.63
GPTQ-T	6.52×10^4	5.55×10^4	2.02×10^7	26.11	24.06	26.22	51.96	49.17	0.02	25.60
TernaryLLM	20.94	45.44	70.46	39.83	28.16	45.54	60.14	**54.30**	29.74	24.66
CAT-Q-T	2.22×10^4	1.79×10^4	2.18×10^7	26.03	24.06	26.64	49.84	49.17	0.00	24.66
PT^2-LLM-T	348.16	424.51	1.48×10^5	28.73	23.72	31.02	52.67	49.33	0.56	23.04
TWLA-T	418.79	486.14	1.51×10^4	30.64	24.49	33.12	54.79	50.36	2.35	24.67
LoopQ-T	1094.40	675.16	5.56×10^4	26.35	22.10	27.15	52.56	51.70	1.24	**27.03**
Megrez	**17.84**	**29.54**	**37.76**	**44.92**	**29.64**	**53.26**	**63.29**	52.09	**35.48**	25.61
Parcae-1.3B										
FP16/BF16	15.41	20.27	14.62	56.12	36.18	64.82	72.25	58.80	44.21	25.61
GPTQ-T	1.27×10^7	2.97×10^7	9.73×10^{10}	25.88	23.72	27.48	51.96	50.83	0.00	24.29
TernaryLLM	29.66	82.84	616.31	33.92	24.49	41.37	56.64	53.83	14.36	24.93
CAT-Q-T	1.39×10^{19}	1.29×10^{19}	2.95×10^{27}	26.23	26.96	27.02	50.98	49.72	0.06	22.95
PT^2-LLM-T	609.63	795.34	4.13×10^4	28.29	24.92	31.61	51.03	50.75	3.84	24.72
TWLA-T	38.91	70.86	138.84	34.89	24.49	47.48	62.19	**54.46**	19.64	24.52
LoopQ-T	1499.37	1264.15	4705.18	29.80	21.42	36.03	56.04	51.22	5.47	**26.34**
Megrez	**20.13**	**28.59**	**48.24**	**44.00**	**28.98**	**55.43**	**66.59**	53.78	**28.62**	24.17
LoopFormer 3×8										
FP16/BF16	27.27	29.86	26.32	32.35	24.74	41.08	63.49	51.78	37.82	24.22
GPTQ-T	7191.51	7452.18	1.65×10^6	26.06	23.46	26.77	51.03	50.43	0.00	23.57
TernaryLLM	263.10	338.91	6449.40	27.63	21.67	32.87	57.24	52.33	2.78	22.95
CAT-Q-T	5879.04	5313.48	1.03×10^6	26.23	**24.57**	26.47	52.29	51.07	0.00	23.39
PT^2-LLM-T	164.18	164.15	783.30	28.67	20.31	34.89	57.51	49.49	11.97	**25.58**
TWLA-T	54.08	64.36	116.48	29.79	21.42	34.43	58.16	50.83	**26.96**	24.75
LoopQ-T	478.50	310.95	5571.61	27.91	20.82	32.24	55.71	51.93	4.89	23.33
Megrez	**36.20**	**43.53**	**74.21**	**30.14**	22.21	**37.49**	**58.87**	**52.51**	26.89	24.61

Final-state calibration. Parcae reaches WT2/C4 20.13/28.59, compared with 38.91/70.86 for its TWLA-T initializer: reductions of 48.3%/59.6%. The complete recipe improves eight of ten metrics over that initializer, including HellaSwag by 9.11 percentage points; WinoGrande and MMLU favor the initializer. For Parcae, Megrez and KL-only both use Adam for log scales and FP64 safeguards for assignment distances (Appendix D). CAT-Q-T is a restricted reimplementation. On LoopFormer, WT2/C4 reaches 36.20/43.53, improving on the recorded TWLA-T row by 33.1%/32.4%. Appendix D also reports a matched-host BF16 parent reload, which retains that direction of improvement separately from the historical table row. Appendix C provides paired seed results and descriptive variation. The following matched controls quantify the contribution of target supervision within each initialized calibration pipeline.

4.3 Target construction and component analysis

What can change while the readout stays fixed? We compare target families on the *same* teacher and student states: 32 fixed windows before student calibration and after 256 parameter updates for each model. Inputs, token positions and the all-state normalization are identical across modes within a snapshot (Appendix G). Figure 4 shows that Megrez removes a substantial part of the mismatch while preserving the readout constraints. Averaging the relative reduction in squared target error across windows gives 20.2%/21.6% on Ouro, 35.1%/36.6% on Parcae and 47.3%/55.5% on LoopFormer before/after calibration, respectively. These comparisons measure target construction on an

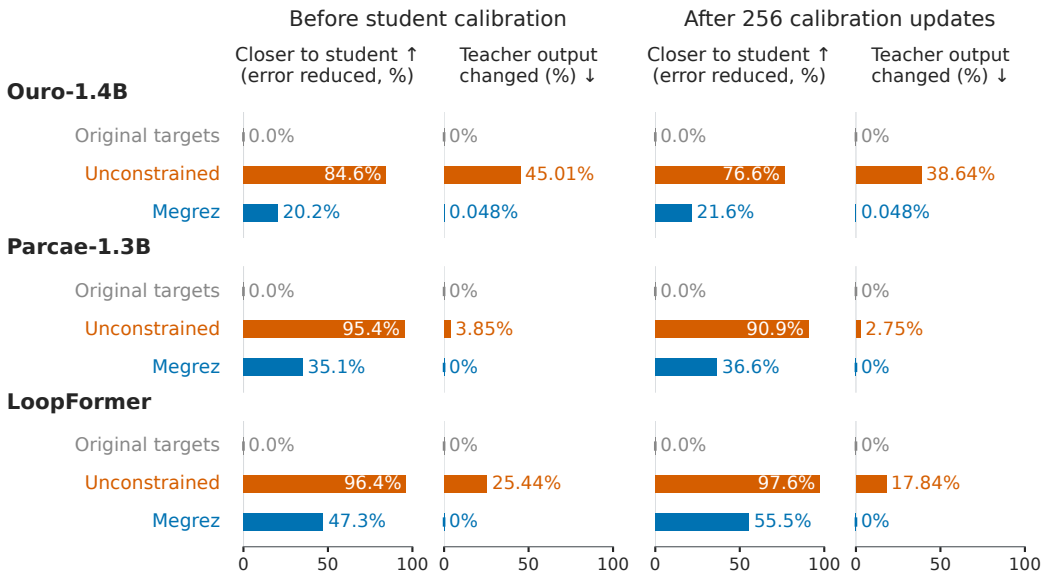

Figure 4: Megrez reduces target error while preserving teacher output. Unconstrained alignment reduces error further but changes output. Columns use fixed student snapshots before calibration and after 256 updates; reconstruction is applied in both. Within each snapshot, variants share teacher/student states and 32 windows without updating weights. Paired bars show mean per-window squared-error reduction relative to original teacher targets and relative output-logit L_2 change in deployment precision, both on 0–100% scales. Output zeros indicate exact native-logit equality; Ouro's 0.048% Megrez residual is FP16 rounding drift. Full Ouro controls appear in Table 12.

Table 2: Matched controls for Megrez: lower-loss method (mean gap).

A. PPL vs. KL-only	WikiText-2	C4
Ouro-1.4B	Megrez (0.0115)	KL-only (0.0138)
Parcae-1.3B	Megrez (0.0123)	KL-only (0.0111)
LoopFormer 3×8	KL-only (0.0132)	Megrez (0.0023)

B. C4 NLL ($\times 10^{-3}$)	KL-only	Fixed targets	Free alignment
Ouro-1.4B	KL-only (6.465)	Fixed (4.657)	Free (7.396)
Parcae-1.3B	KL-only (18.979)	Fixed (25.613)	Free (35.294)
LoopFormer 3×8	Megrez (0.241)	Megrez (0.014)	Megrez (0.403)

Notes. Margins are absolute mean differences, not significance claims. All runs use 256 updates. A: seeds 20260909–11 (Appendix C.1). B: 256 C4 documents; two new Ouro/Parcae seeds (20260926/27); three separate matched LoopFormer reruns (20260909–11) on previously reported documents (Appendix H.3). Fixed/Free: fixed-state/unconstrained-alignment targets.

unchanged student. A supplementary Ouro-2.6B check removes 16.3–17.4% of the mismatch across two frozen snapshots while preserving the same readout constraints (Appendix I).

Unconstrained alignment reduces distance further but changes the readout. On Ouro, preserving only the weighted sum allows altered gates to change logits; both constraints reduce ideal relative logit error below 1.6×10^{-14}, with mean FP16 drift about 4.8×10^{-4} (Table 12). Restoring the terminal state preserves native BF16 logits elementwise for Parcae/LoopFormer in this diagnostic; Appendix B distinguishes the training solvers. Parcae's training traces reduce cumulative alignment error by 33.0–46.8% (Appendix C.2).

Contribution of the auxiliary objective. The KL-only control removes the auxiliary term while matching the initializer, three seeds, data order, numerical policy and 256 updates under the original refresh-eight, decaying-coefficient schedule (Table 2A). Megrez lowers mean WT2 on Ouro/Parcae and raises it on LoopFormer; C4 reverses these directions. Parcae's WT2 difference is -0.0123 (paired sample SD 0.5074). Both improve on their initializers; Table 4 gives all seed pairs.

4.4 Target families and optimization schedule

Across six Ouro target families with matched training and native four-exit inference, both constraints give the lowest mean WT2 and highest mean HellaSwag; fixed multiple exits give the lowest mean C4. Both constraints beat free alignment on WT2/C4 in both seeds (mean PPL reductions 0.0165/0.0218), while HellaSwag reverses sign. Target-distance rankings differ from predictive rankings; fixed-multiple versus fixed-final also changes loss normalization (Appendix F).

Neither per-update reconstruction nor a constant auxiliary coefficient beats both original Megrez and KL-only in the 64-document screen; the prespecified rule retains the original schedule. On 256 C4 documents (Table 2B), all three controls lower mean NLL on Ouro/Parcae; free alignment wins in both new seeds. Parcae's fixed/free targets also lead all three component benchmarks in both seeds. Separate LoopFormer reruns reuse reported documents: means favor Megrez, but fixed/free differences change sign across pairs. These controls do not establish a stable predictive benefit from target reconstruction (Appendix H).

Recurrent depth. Figure 5 contrasts two frozen-weight responses. Ouro's WT2 perplexity falls from 172.4 at two calls to 17.8 at four, then stays near 17.9 at eight; its NLL gaps also plateau. LoopFormer degrades from 36.2 at eight calls to 276.2 at sixteen even with total time fixed at $T = 1$. Its modest WT2 gap to FP therefore does not imply good absolute quality. Extra recurrence offers no consistent improvement without retraining. Appendix C.3 retains Parcae, both LoopFormer time conventions and every seed-level result (Table 6).

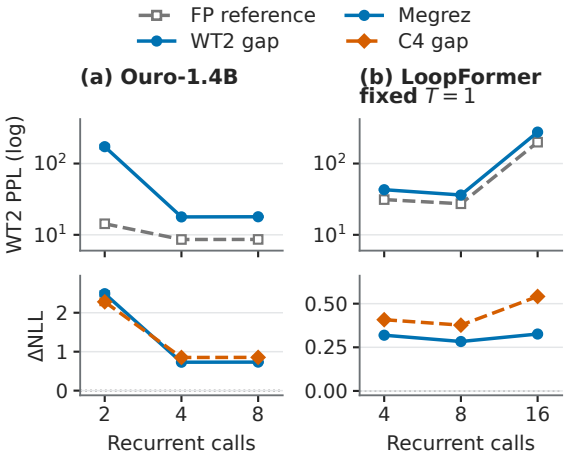

Figure 5: Frozen depth response: WT2 perplexity (top) and Megrez minus same-depth FP NLL (bottom). Means ± sample SD over three seeds; no retraining. The fixed-T diagnostic followed the fixed-step sweep. Full curves: Figure 6.

5 Limitations

Exact target construction does not guarantee predictive gains from fitting shared ternary weights. Implementations use rank thresholds, normalization floors and cached targets. LoopFormer's initializer uses test results; its added controls reuse reported C4 documents. External recipes differ in initialization and budget. Document intervals exclude training-seed variation. Complete calibration-cost ancestry, packed-kernel execution and full historical training reproduction remain unverified.

6 Conclusion

Megrez reconstructs intermediate supervision with an exact orthogonal alignment that preserves teacher outputs and normalized state geometry. The complete calibration framework improves WT2/C4 perplexity over the evaluated initializers and adapted recipes across three architectures. Matched controls show that improved target alignment does not consistently improve prediction, supporting separate evaluation of target geometry and predictive utility.

AI USE STATEMENT

AI assistants were used for code development and inspection, organizing experimental records, developing and checking mathematical arguments, preparing figures and build scripts, and drafting and revising manuscript text. Reported measurements are drawn from archived completed evaluations. AI-generated estimates are not used as experimental results. The authors are responsible for the scientific claims and the final submission.

ETHICS STATEMENT

This study evaluates existing language-model checkpoints and text benchmarks; it introduces no newly collected participant data or human-subject study. The results concern predictive quality and resource use, and do not evaluate whether quantization preserves safety-related behavior. Third-party models and datasets remain subject to their original access and license conditions.

REPRODUCIBILITY STATEMENT

Appendix E provides the complete target-construction proofs and algorithm. Appendices B and C specify the implemented recipes, evaluation protocols, all reported seed pairs and deployment measurements. Appendix D identifies numerical successors, restricted adaptations and provenance gaps. The source records distinguish fixed final endpoints from historical and unfinished runs; missing measurements are explicitly marked. The anonymous source package includes a NumPy target-operator reference, CPU algebra checks, paired measurements and plotting code. Using an independently extracted training archive, we completed one 256-update Parcae calibration and native reload/evaluation (Appendix A). The reproduction record identifies the code and final checkpoint used in this run. Initializer training and the full historical experiment lineage were not reproduced. Appendix H specifies the development selection, fresh-document confirmation, separate H100 training reproduction and repeated evaluation on RTX 4090 D. It also separates the added LoopFormer target-family reruns on previously reported documents from the original confirmation cohorts. The source supplies document/token identities, reconstruction scripts, per-document losses and the statistical calculations. Appendix I adds the Ouro-2.6B scale check, with window-level measurements, native evaluation losses, timing samples and per-file storage counts in the accompanying data.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 100213–100240. Curran Associates, Inc., 2024. doi: 10.52202/079017-3180. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b5b939436789f76f08b9d0da5e81af7c-Paper-Conference.pdf.

Prajjwal Bhattarai, Mohammad Amjad, Dmytro Zhylko, and Tuka Alhanai. Knowledge distillation through geometry-aware representational alignment, 2025. URL https://arxiv.org/abs/2509.25253.

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

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. Quip: 2-bit quantization of large language models with guarantees. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 4396–4429. Curran Associates, Inc., 2023. doi: 10.52202/075280-0196. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/0df38cd13520747e1e64e5b123a78ef8-Paper-Conference.pdf.

Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11933–11942, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chen_Knowledge_Distillation_With_the_Reused_Teacher_Classifier_CVPR_2022_paper.html.

Tianqi Chen, Zhe Li, Weixiang Xu, Zeyu Zhu, Dong Li, Lu Tian, Emad Barsoum, Peisong Wang, and Jian Cheng. Ternaryllm: Ternarized large language model, 2024. URL https://arxiv.org/abs/2406.07177.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers, 2019. URL https://arxiv.org/abs/1807.03819.

Rui Fang, Hsi-Wen Chen, and Ming-Syan Chen. Loopq: Quantization for recursive transformers, 2026. URL https://arxiv.org/abs/2605.16343.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach, 2025. URL https://arxiv.org/abs/2502.05171.

Sang Il Han. Teacher supervision over representation equivalence classes, 2026. URL https://arxiv.org/abs/2607.03572.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. Loopformer: Elastic-depth looped transformers for latent reasoning via shortcut modulation, 2026. URL https://arxiv.org/abs/2602.11451.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations, 2025. URL https://arxiv.org/abs/2405.16406.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits, 2024. URL https://arxiv.org/abs/2402.17764.

Vladimir Malinovskii, Denis Mazur, Ivan Ilin, Denis Kuznedelev, Konstantin Burlachenko, Kai Yi, Dan Alistarh, and Peter Richtarik. Pv-tuning: Beyond straight-through estimation for extreme llm compression, 2024. URL https://arxiv.org/abs/2405.14852.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Roy Miles, Ismail Elezi, and Jiankang Deng. VkD: Improving knowledge distillation using orthogonal projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15720–15730, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Miles_VkD_Improving_Knowledge_Distillation_using_Orthogonal_Projections_CVPR_2024_paper.html.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3967–3976, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Park_Relational_Knowledge_Distillation_CVPR_2019_paper.html.

Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, and Daniel Y. Fu. Parcae: Scaling laws for stable looped language models, 2026. URL https://arxiv.org/abs/2604.12946.

Chengyao Qian, Trung Le, and Mehrtash Harandi. A good teacher adapts their knowledge for distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1239–1248, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Qian_A_Good_Teacher_Adapts_Their_Knowledge_for_Distillation_ICCV_2025_paper.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015. URL https://arxiv.org/abs/1412.6550.

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

Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451. URL https://doi.org/10.1007/BF02289451.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models, 2024. URL https://arxiv.org/abs/2308.13137.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4323–4332, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1441. URL https://aclanthology.org/D19-1441/.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1365–1374, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Tung_Similarity-Preserving_Knowledge_Distillation_ICCV_2019_paper.html.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, and Anbang Yao. Attend to your own thoughts: Breaking the barrier for post-training quantization of reasoning llms through the lens of 1.58-bit quantization, 2026a. URL https://arxiv.org/abs/2608.01078.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, and Anbang Yao. Cat-q: Cost-efficient and accurate ternary quantization for llms, 2026b. URL https://arxiv.org/abs/2606.26650.

Xianglong Yan, Chengzhu Bao, Zhiteng Li, Tianao Zhang, Kaicheng Yang, Haotong Qin, Ruobing Xie, Xingwu Sun, and Yulun Zhang. Pt^2-llm: Post-training ternarization for large language models, 2026. URL https://arxiv.org/abs/2510.03267.

Wanqi Yang and Shiwei Liu. Flashloop: Fast and memory-efficient looped transformers via lazy updates, 2026. URL https://arxiv.org/abs/2609.29812.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1904.00962.

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

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

Zhixiong Zhao, Zukang Xu, Zhixuan Chen, Xing Hu, Zhe Jiang, and Dawei Yang. Twla: Achieving ternary weights and low-bit activations for llms via post-training quantization, 2026. URL https://arxiv.org/abs/2606.13054.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models, 2026. URL https://arxiv.org/abs/2510.25741.

A Reproducibility and Evaluation Protocol

Models and calibration. The checkpoints are `ByteDance/Ouro-1.4B`, `SandyResearch/parcae-1.3b`, and `armenjeddi/LoopFormer-3block-8iterations`, with four, eight and eight recurrent calls. Megrez uses 128 calibration windows, 256 updates and seeds 20260909–20260911. Means include all three final checkpoints. KL-only pairs share the initializer, data order and numerical policy.

Evaluation. The ten metrics are WT2, C4 and LAMBADA-OpenAI perplexity; HellaSwag, ARC-Challenge, ARC-Easy, PIQA, WinoGrande, LAMBADA-OpenAI and MMLU accuracy. HellaSwag, ARC and PIQA use length-normalized accuracy. MMLU uses all 14,042 test examples in 57 subjects, the first five development examples per subject as shots, and size-weighted aggregation (lm-eval 0.4.9, `cais/mmlu`). Other tasks use full zero-shot splits. WT2 uses non-overlapping test windows: 148 for Ouro and 151 for Parcae at 2048 tokens, and 280 for LoopFormer at 1024 tokens (Merity et al., 2016). C4 uses 256 frozen evaluation chunks at the model's context length. Calibration and test arrays are distinct.

Uncertainty. Megrez's C4 ranges are 29.4815–29.6284 (Ouro), 27.9378–29.4688 (Parcae) and 43.4866–43.5647 (LoopFormer). Each seed's WT2/C4 is below the reported TernaryLLM mean on Ouro or the TWLA-T endpoint on the final-state models. These comparisons use a common reference. Seed SDs describe variation on fixed evaluation sets; Appendix H reports paired intervals for new documents conditional on the trained models.

A.1 Reproduction scope

An independent Parcae reproduction (seed 20260926) completed 256 updates and exported 144 projections. Reloading reproduced all 64 development document losses exactly and evaluated the 256 confirmation documents. It used the supplied calibrated initializer and licensed assets. Teacher pretraining, initializer production and a fresh software installation were not repeated; the supplementary code does not include every historical checkpoint, token sequence or environment.

B Implementation Details

Embeddings, readout components, normalization parameters and fixed initializer transforms remain unchanged during calibration. Benchmarks evaluate the exported hard weights.

Table 3: Implemented Megrez instances. Calibration windows, token sampling and forward precision are model-specific; all main endpoints use 256 updates and the three registered seeds.

	Ouro-1.4B	Parcae-1.3B	LoopFormer
Physical projections	168	144	12
Supervised recurrent states	4	8	8
Calibration windows	128	128	128
Tokens per window	2048	2048	1024
Auxiliary token stride	32	16	16
Sampled positions per window	64	128	64
Target refresh interval	8	8	8
Hard deployment precision	FP16	BF16	BF16
Gradient forward precision	FP32	BF16	BF16
Initializer	native-PV	TWLA-T	TWLA-T

Schedules and updates. The initial learning rates are $3 \cdot 10^{-4}$ for continuous group parameters and $3 \cdot 10^{-3}$ for assignment targets, with cosine decay to one tenth of those values. The auxiliary coefficient decreases linearly from 0.03 to zero. Target refresh uses the next eight windows in the seeded permutation of the 128-window calibration set. No gradient passes through cached targets. Ouro uses an FP32 gradient proxy and FP16 state capture and evaluation. LoopFormer and the current Parcae cohort retain native BF16 forward behavior with FP32 optimization parameters. The

Parcae successor changes every log scale to Adam, retaining LAMB for other parameters and an FP64 guard for unsafe assignment distances. This revision applies symmetrically to both arms.

Numerical target construction. The Ouro and final-state solvers use FP64 thin SVDs and discard singular directions below $\max(m, n)\epsilon_{64}\sigma_{\max}$ when constructing a basis for an $m \times n$ matrix. This numerical rank choice is distinct from exact rank. Ouro checks gate-score error below 10^{-8}, survival-mass error below 10^{-8}, aggregate relative error below 10^{-9} and normalized-length relative error below 10^{-9} inside its target solver. These are operator checks, not bitwise FP16 readout guarantees.

Final-state target construction divides raw states by RMS floored at 10^{-12}; the differentiable student residual uses a 10^{-6} floor. The ideal unit-RMS equations presume those floors are inactive. Targets are renormalized before caching in FP32, and their squared norm supplies the implemented residual denominator. For Parcae, the final raw state is copied back from the teacher only after checking normalized roundoff below 10^{-9}; this preserves the anchor exactly but does not make every finite-precision Gram entry exact. LoopFormer's training solver retains a floating-point residual; its target-operator error of at most $9 \cdot 10^{-16}$ does not bound the subsequent BF16 readout error.

Independent operator verification. We compare the reduced Ouro and final-state solvers with a full-complement SVD on synthetic FP64 inputs. Generic, repeated-state and single-state cases agree in objective within $2.9 \cdot 10^{-13}$; normalized Gram errors stay below $9.3 \cdot 10^{-14}$.

C Predictive Controls, Depth and Deployment

C.1 Matched auxiliary-loss controls

Table 4: Matched native-depth perplexities for every training seed at update 256. A uses output KL and C adds reconstructed targets. Differences are C minus A. All values come from completed evaluations; no intermediate endpoint is selected.

Model	Seed	A WT2	C WT2	ΔWT2	A C4	C C4	ΔC4
Ouro-1.4B	20260909	17.8789	17.8902	0.0113	29.4924	29.4976	0.0052
Ouro-1.4B	20260910	17.8694	17.8391	-0.0303	29.4671	29.4815	0.0145
Ouro-1.4B	20260911	17.7913	17.7757	-0.0156	29.6068	29.6284	0.0216
Parcae-1.3B	20260909	20.0953	20.6414	0.5461	28.4701	29.4688	0.9987
Parcae-1.3B	20260910	20.1886	19.7433	-0.4453	28.6440	27.9378	-0.7062
Parcae-1.3B	20260911	20.1373	19.9995	-0.1378	28.6369	28.3778	-0.2591
LoopFormer	20260909	36.2612	36.2325	-0.0287	43.5140	43.5288	0.0148
LoopFormer	20260910	36.1775	36.2345	0.0570	43.5826	43.5647	-0.0179
LoopFormer	20260911	36.1292	36.1405	0.0113	43.4902	43.4866	-0.0036

The controls share initialization, 256 updates, data order and numerical policy; only the auxiliary term differs. Ouro improves mean WT2, while C4 and LAMBADA perplexity favor KL-only. Parcae's WT2 differences (Megrez minus KL-only) are $+0.5461$, -0.4453 and -0.1378, with mean -0.0123 and sample SD 0.5074. Its C4 difference is $+0.0111$ with SD 0.8840. LoopFormer's WT2/C4 directions also vary across seeds. The KL-only task means below complement the Megrez means in Table 1, separating each metric's mean effect from variation across the three seed pairs.

Table 5: KL-only internal controls corresponding to the Megrez rows. All three models use arithmetic means of three 256-update seeds. Parcae uses scale-Adam in both objectives (Appendix D).

Model	WT2↓	C4↓	LamP↓	HS↑	ARC-C↑	ARC-E↑	PIQA↑	WG↑	Lam↑	MMLU↑
Ouro-1.4B	17.8465	29.5221	37.6062	44.825	29.721	52.792	63.185	52.013	35.365	25.436
Parcae-1.3B	20.1404	28.5837	49.4698	44.018	28.840	55.191	66.250	53.802	28.217	24.206
LoopFormer	36.1893	43.5290	74.5661	30.177	22.355	37.598	58.959	52.170	26.975	24.624

C.2 TARGET ALIGNMENT DURING CALIBRATION

All three completed Parcae Megrez runs retain 256 target-construction records, each covering 128 sampled positions and eight states. Summing squared residuals before (E_j) and after (E'_j) reconstruction, $100(1 - \sum_j E'_j / \sum_j E_j)$ is 32.97%, 46.78% and 38.15% for seeds 20260909–20260911. All 768 records improve alignment. Windows recur, so these are trajectory observations, not independent test examples. The largest normalized final-state change before roundoff correction is 3.5528×10^{-15}; the raw final state is unchanged after correction. Median weighted-auxiliary/KL loss ratios are 0.0851%, 0.0559% and 0.0789%. Appendix H measures parameter-gradient influence separately.

C.3 RECURRENT DEPTH WITH FIXED WEIGHTS

We evaluate FP and all three Megrez checkpoints at half, native and double the recurrent call count, using the same WT2/C4 arrays, batch one and native precision. Evaluation seed 20260909 is reset for each depth/metric. Ouro and Parcae use A100 PCIe; LoopFormer uses RTX 4090 D, with an FP reference on each physical card. LoopFormer initially fixes $\Delta t = 1/8$; a separate control fixes total time $T = 1$ through $\Delta t = 1/d$. The latter was added after observing degradation away from eight calls and changes both step size and call count. Both schedules agree exactly at eight calls. Table 6 reports every depth measurement. WT2 uses 302,956, 309,097 and 286,440 predicted tokens for Ouro, Parcae and LoopFormer, respectively; C4 uses 524,032 for Ouro/Parcae and 261,888 for LoopFormer. Perplexity is the exponential of the mean token NLL. Results at the original depth are kept separate from Table 1 because the evaluation hardware and software environments can differ. Parcae's same-A100 seed 20260910 reproduces its historical NLL exactly; the two originally RTX-evaluated seeds differ in the A100 cohort. The source of these differences has not been isolated. The sweep compares FP and Megrez; other baselines were not evaluated at every depth.

Table 6: Perplexity at half, original and double recurrent depth, with fixed weights (lower is better). All 108 measurements are shown, rounded to two decimals. S09/S10/S11 denote Megrez seeds 20260909/10/11; d_0 is the original call count. LoopFormer uses two RTX 4090 D cards: A for S09 and B for S10/S11, each with its own floating-point (FP) reference. The two FP references agree exactly. Both time schedules also agree exactly at $d_0 = 8$. No depth-specific retraining or model selection is performed.

	WT2			C4		
Model / seed	$d_0/2$	d_0	$2d_0$	$d_0/2$	d_0	$2d_0$
Ouro-1.4B ($d_0 = 4$; A100 PCIe)						
FP	14.29	8.60	8.62	17.99	12.61	12.62
Megrez S09	161.71	17.89	17.94	162.01	29.50	29.56
Megrez S10	178.87	17.84	17.95	177.31	29.48	29.60
Megrez S11	176.63	17.78	17.87	189.41	29.63	29.73
Parcae-1.3B ($d_0 = 8$; A100 PCIe)						
FP	15.93	15.41	15.70	21.18	20.28	19.47
Megrez S09	23.72	20.77	20.79	32.21	29.52	29.25
Megrez S10	20.80	19.74	19.76	29.39	27.94	27.94
Megrez S11	21.57	19.98	20.02	30.29	28.34	28.23
LoopFormer ($d_0 = 8$; fixed step $\Delta t = 1/8$)						
FP (card A)	2034.13	27.27	49791.95	1328.78	29.86	45145.77
FP (card B)	2034.13	27.27	49791.95	1328.78	29.86	45145.77
Megrez S09	2199.10	36.23	44918.62	1467.21	43.53	33887.62
Megrez S10	2154.61	36.23	44468.36	1433.03	43.56	32396.36
Megrez S11	2192.67	36.14	44207.74	1467.80	43.49	32125.14
LoopFormer ($d_0 = 8$; fixed total time $T = 1$)						
FP (card A)	31.21	27.27	199.32	32.59	29.86	158.06
FP (card B)	31.21	27.27	199.32	32.59	29.86	158.06
Megrez S09	43.08	36.23	277.05	49.05	43.53	271.52
Megrez S10	42.96	36.23	276.48	49.04	43.56	271.77
Megrez S11	42.81	36.14	274.96	48.97	43.49	271.48

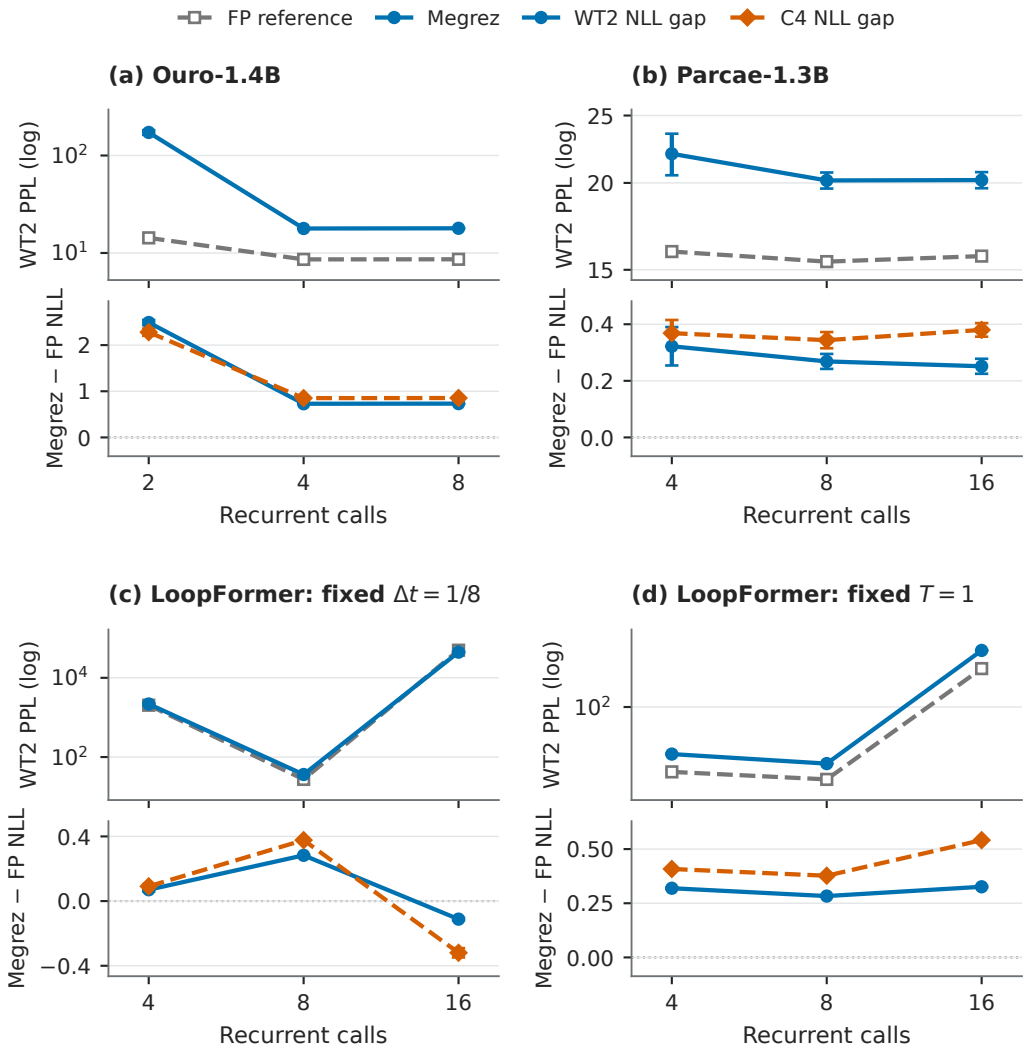

Figure 6: Frozen-checkpoint depth behavior. Within each panel, top: WT2 perplexity on a log scale; bottom: Megrez minus same-depth FP NLL on WT2 and C4. Points and bars show means and sample standard deviations across three Megrez seeds. Both LoopFormer time schedules are retained; the fixed-horizon diagnostic was added after the fixed-step sweep. Appendix C.3 specifies the protocol; Table 6 reports all seed-level results.

C.4 Representation and recipe fairness

Table 7: Representation and retained components of the Ouro-1.4B adaptations. Group sizes, grids and additional transforms differ. TernaryLLM uses a restricted calibration budget; CAT-Q-T is a paper reimplementation. Megrez's target transformations are discarded after calibration. Fixed initializer transforms remain part of its deployment.

Method	Weights	Act.	Group	Scale/offset	Residual	Status
FP16/BF16	full precision	16	–	–	No	Reference
GPTQ-T	ternary adaptation	16	128	yes	No	Adapted
TernaryLLM	ternary; DLT+OFF	16	128	yes	No	Adapted, restricted
CAT-Q-T	reimplemented ternary	16	128	yes	No	Adapted
PT^2-LLM-T	ternary; GPTAQ retained	16	128	yes	No	Adapted
TWLA-T	ternary; one-plane KOTMS	16	128	yes	No	Adapted
LoopQ-T	ternary; CTA+SLT	16	32	scale only	CTA	Adapted
Megrez	shared affine ternary	16	128	yes	No	Ours

C.5 Efficiency and deployment

Megrez adds no inference module to its initialized student. The teacher, reconstructed targets and optimizer are calibration assets. The Ouro-1.4B active payload of 818.431 MB counts packed ternary projections, FP32 grids, retained initializer transforms, protected FP16 tensors and column maps once, despite recurrent reuse. It excludes the loader's original checkpoint/tokenizer and dense runtime caches; it is neither a standalone package size nor a runtime-memory measurement.

Recorded final-stage process hours are 2.0438/1.6977/1.4424 for TernaryLLM and 1.4151/1.8935/3.8688 for Megrez, on mixed H100/A100 hardware. Peak reserved memory is 29.904–29.947 GiB and 31.518 GiB, respectively. These stages cover setup, calibration, development scoring and export; upstream search, initializer production, failures and benchmark evaluation are excluded. Mixed hardware and contention prevent a controlled training-speed comparison.

Table 8: Native dense forward cost on A100 PCIe. Megrez columns give all three seed means in order 20260909/10/11; each mean uses 40 forwards. Memory is peak allocated/reserved GiB (identical across the three Megrez seeds).

Model	Tokens	FP mean (ms)	Megrez means (ms)	Megrez GiB
Ouro-1.4B	2048	241.16	299.29 / 299.59 / 298.53	3.485 / 5.357
Parcae-1.3B	2048	108.80	148.34 / 155.14 / 148.27	3.209 / 3.688
LoopFormer	1024	20.19	27.01 / 26.74 / 26.72	0.867 / 1.186

Timing uses an otherwise idle A100 PCIe, batch one, native calls and precision, eight warm-ups and 40 forwards per endpoint, cycling the first four frozen WT2 windows. Inputs and weights are resident. Synchronized wall time covers only the forward pass; loading, transfer, checks, KV caching and autoregressive decoding are excluded. The host remains shared. Parcae resets seed 20260909 for its stochastic initial state. Allocated memory counts live tensors; reserved memory includes retained allocator cache. All three Megrez implementations use dense FP16/BF16 caches and run slower than their floating-point references; packed ternary kernels are not measured. Forward timings for the final Ouro/Parcae LoopQ models remain unavailable.

D Initialization and Adaptation Boundaries

Initialization. Ouro starts from an optimized native-PV affine deployment with two trained projection overlays (WT2/C4 29.37/33.44). Parcae and LoopFormer start from single-plane TWLA-T and retain its fixed transforms. Initializer calibration belongs to the complete recipe. LoopFormer losslessly relabels 43 negative-scale groups and then updates the same 12 shared projections in both arms. A same-host BF16 reload gives parent WT2/C4 54.16/64.47; one seed's KL-only and Megrez continuations give 36.26/43.51 and 36.23/43.53. The older RTN-initialized Megrez endpoint gave 463.87/652.64 and motivated the initializer change. This choice was informed by test results, so the later controls evaluate the chosen recipe rather than test-unseen selection.

Parcae numerical version. In the original recipe, seed 20260910 with the auxiliary term failed at update 117 after 116 completed updates through nonfinite assignment arithmetic. Seed 20260909's WT2/C4 favored Megrez (18.63/27.86 versus 20.09/34.31), while seed 20260911 favored KL-only (17.92/28.13 versus 23.52/41.85 for Megrez); the latter also favored KL-only on LAMBADA perplexity and six of seven accuracies. No surviving-seed mean replaces this incomplete cohort. The successor applies Adam to all log scales, LAMB elsewhere and guarded FP64 assignment distances symmetrically to both arms. All six fresh runs complete 256 updates. A rule fixed before the successor test results requires all three pairs and replaces the old main row irrespective of score direction; versions are not pooled. Tables 4 and 5 report the successor's paired perplexities and complete control means.

TernaryLLM and PT^2-LLM. Ouro TernaryLLM retains DLT and OFF with the restricted 128-window, 256-update budget. PT^2-LLM-T retains GPTAQ on Ouro; final-state adaptations omit it. LoopFormer's corrected PT^2 row restores the stored SSR physical-column permutation without refitting its weights and uses complete new evaluation.

TWLA and LoopQ. TWLA-T uses single-plane fits. LoopQ-T uses CTA, selective per-loop copies, group 32, 1024 documents and 3840 updates. Ouro additionally needs documented numerical adaptations: a 2^{-14} floor for transform diagonals, a detached power-of-two scale for finite weights that overflow FP16 casting, and extended intermediate range for attention-output/MLP-down projections immediately followed by RMSNorm. The latter carries y/s and changes epsilon to ϵ/s^2, preserving the real-valued normalization expression. Ordinary finite paths, parameters and calibration settings are retained.

CAT-Q and execution precision. CAT-Q-T is a paper reimplementation with literal final-soft-gradient hardening. Its Parcae fit completes all 27 reconstruction windows and produces finite but very large perplexities, which remain reported. The adaptations are not equivalent to the original authors' full recipes. Ouro's retained GPTQ/CAT-Q/PT^2/TWLA loaders use BF16, while some artifact labels say FP16; historical quality logs do not bind the actual execution dtype completely. The later runtime cohort directly checks BF16.

E Target Construction: Assumptions, Proofs and Algorithm

The following results concern Equation 1 in exact arithmetic. Teacher states and the student snapshot are fixed during each solve, with hidden dimension d and K supervised states. Ouro requires a frozen invertible diagonal gain; final-state normalization assumes nonzero states. Readout parameters are fixed. Numerical guards, rank thresholds and the precision of the training forward are separate implementation choices.

E.1 Preservation

Let $\widetilde{Y} = RY$. Orthogonality gives $\widetilde{Y}^\top \widetilde{Y} = Y^\top R^\top RY = Y^\top Y$. All normalized-state norms and pairwise inner products are preserved. For a fixed direction b with $Rb = b$, multiplying by $R^\top$ yields $R^\top b = b$. For Ouro, use $b = \nu = Dv$:

$$v^\top \widetilde{h}_t^T + \beta = v^\top DRy_t + \beta = \nu^\top Ry_t + \beta = \nu^\top y_t + \beta.$$

Sigmoid scores and survival products are therefore identical. The constraint $R\mu = \mu$ proves Equation 4; a fixed linear head maps identical aggregates to identical logits. A shared head bias is preserved because $\sum_t \alpha_t = 1$. The result holds for any rank of W_o. Fixing the aggregate supplies sufficient constraints for the orthogonal target family.

Why a fixed aggregate alone is insufficient. Consider a two-exit scalar example with $D = 1$, $y_1 = 1$, $y_2 = -1$, $v = 1$, $\beta = -1$ and an identity output head. The teacher has $\alpha_1 = \sigma(0) = 1/2$, $\alpha_2 = 1/2$ and $\mu = 0$. The orthogonal transformation $R = -1$ fixes this aggregate and preserves the Gram matrix. Recomputing its gate gives $\widetilde{\alpha}_1 = \sigma(-2)$, however, so its actual readout becomes $1 - 2\sigma(-2) \neq 0$. This analytic example separates preservation of the old-mass aggregate from preservation of the state-dependent readout. Fixing the gate direction excludes this transformation.

For the final-state instance, $Ry_K = y_K$ implies $\widetilde{h}_K^T = \rho(h_K^T)y_K = h_K^T$. Any fixed deterministic downstream readout therefore receives the same complete final-state sequence, including when it contains further Transformer layers. Unselected token states are unchanged and selected final states are fixed. Restoring per-state scalar RMS also preserves raw pairwise inner products within each token for these instances. Ouro instead restores an anisotropic diagonal gain; only its pre-gain Gram matrix is guaranteed.

E.2 Existence and the alignment bound

Figure 2 illustrates the remaining freedom with a three-dimensional final-anchor example. A common orthogonal map rotates the intermediate targets while fixing the final state and every state's coordinate along its direction. The full objective includes both rotations and reflections in the free plane.

The set $\mathcal{G}_B$ is a closed subset of the compact orthogonal group and contains the identity. A continuous squared-distance objective attains its minimum. Evaluating that objective at the identity gives the non-increase bound in Proposition 2 for the fixed student snapshot.

Figure 3 gives an explicit example of this reduction.

E.3 Why at most $2K$ active dimensions suffice

Set $F = \operatorname{span}(B)$ and $P = I - BB^\top$. An orthogonal map fixing F pointwise leaves $F^\perp$ invariant. The objective decomposes as

$$\|RY - Z\|_F^2 = \|BB^\top(Y - Z)\|_F^2 + \|R(PY) - PZ\|_F^2.$$

The first term is constant. The second is a constant minus $2\operatorname{tr}((PZ)^\top R(PY))$. Let $S = \operatorname{span}[PY, PZ] \subseteq F^\perp$, with orthonormal basis U and $r = \dim S \le 2K$. Set $X = U^\top PY$, $T = U^\top PZ$ and write $TX^\top = L\Sigma V^\top$.

For any feasible R, $U^\top RU$ is a contraction. Its trace pairing with $TX^\top$ is at most $\|TX^\top\|_* = \sum_j \Sigma_{jj}$. The orthogonal matrix $O = LV^\top$ attains this upper bound. Extending O by the identity on $S^\perp$ gives $R = I + U(O - I_r)U^\top$, which is orthogonal and fixes F. Thus an optimizer exists acting only in S; allowing motion outside S cannot improve the optimum. Zero singular values permit nonunique orthogonal completions without changing the objective. If $r = 0$, use the identity. No determinant-one restriction is imposed.

Thin basis construction, forming the $r \times r$ cross-covariance and its SVD cost $O(dK^2 + K^3)$ when the fixed subspace has dimension at most K. Operator workspace is $O(dK + K^2)$, excluding model activations, cached logits and optimizer state.

E.4 Optimal residual and strict improvement

Let $M = TX^\top$ and write $E^\star = \|R^\star Y - Z\|_F^2$. The same Procrustes solution gives

$$\begin{aligned} E^\star &= \|B^\top(Y - Z)\|_F^2 + \|X\|_F^2 + \|T\|_F^2 - 2\|M\|_*, \\ \|Y - Z\|_F^2 - E^\star &= 2\big(\|M\|_* - \operatorname{tr} M\big). \end{aligned} \tag{8}$$

Here $\|M\|_*$ is the sum of singular values. The identity is optimal exactly when M is symmetric positive semidefinite, including $M = 0$. Otherwise the target distance strictly decreases.

The reduced objective is

$$\|B^\top(Y - Z)\|_F^2 + \|X\|_F^2 + \|T\|_F^2 - 2\operatorname{tr}(M^\top O), \qquad M = TX^\top.$$

Its maximum trace term is $\|M\|_*$, whereas the identity gives $\operatorname{tr} M$. Subtracting yields Equation 8. For completeness, write $M = L\Sigma V^\top$. Then $\operatorname{tr} M = \sum_j \Sigma_{jj}(V^\top L)_{jj} \le \sum_j \Sigma_{jj}$. Equality requires $(V^\top L)_{jj} = 1$ for every positive singular value. Since $V^\top L$ is orthogonal, this implies $Le_j = Ve_j$ for these directions and hence $M = V\Sigma V^\top$, a symmetric positive semidefinite matrix. Conversely, such a matrix has trace equal to its nuclear norm. Zero singular values impose no additional restriction. Thus identity optimality is equivalent to this condition, and all other cases have strictly positive optimal gain.

E.5 CALIBRATION ALGORITHM

1. Initialize shared ternary codes, scales and offsets from the frozen model-specific initializer. Retain recorded fixed transforms and protected native tensors. Fix the calibration order and 256-update budget.
2. Before each block of eight updates, capture teacher and current hard student states for the next eight windows. For each sampled token, form the architecture-specific Y, Z, B, solve Equation 5, and cache detached targets and teacher output logits.
3. For each window in the block, run the student's recurrent forward, compute Equation 7, and update continuous parameters and discrete assignments. Retain shared weights across calls. Do not differentiate through reconstruction or substitute targets into recurrence.
4. Export the hard endpoint after the final update and score its native deployment forward. Discard teacher states, target cache, target transformations and optimizer state.

The main recipes use $\eta_k/\eta_1 = 0.1+0.9(1+\cos(\pi(k-1)/255))/2$ and $\lambda_k = 0.03(1-(k-1)/255)$ for $k = 1, \ldots, 256$. The KL-only control sets $\lambda_k = 0$. Reconstruction is optimal for the snapshot used at refresh. Weight updates then change the student under fixed cached targets; the next refresh solves the corresponding new alignment problem. The operator guarantees apply to each solve, without a convergence guarantee for the discrete calibration procedure.

F TARGET-FAMILY ABLATION

All six Ouro variants share the initializer, 128 calibration windows of 2048 tokens, optimizer, 256 updates, eight-update refresh and a coefficient decaying from 0.03 to zero. The fixed-final variant supervises the final state with its own state-energy denominator; the others use multiple states. Inference always retains the native four-exit reader. The comparison thus changes normalization as well as state coverage for fixed-final versus fixed-multiple targets. Both retained seeds (20260910/11) train on A100 PCIe and are evaluated on A100 SXM. The third seed was omitted before final quality results were available under the deadline amendment; its earlier qualification and failed training attempt are included in the costs below.

All 12 final endpoints use 148 WT2 windows, 256 C4 windows and 10,042 zero-shot HellaSwag examples. Quality uses native FP16; training uses the registered FP32 gradient proxy. Final diagnostics use 64 sampled positions in each of 128 calibration windows, FP16 state capture and FP64 target algebra. They describe different trained students; the same-snapshot study in Appendix G supplies the controlled geometric comparison.

Table 9: Matched Ouro-1.4B ablation (two seeds). Panel A: quality mean ± sample SD; HellaSwag normalized accuracy in percent. Panel B: mean target distance and maximum gate/actual aggregate-logit errors over calibration windows and seeds (FP64 target algebra).

Panel A: quality

Target family	WT2↓	C4↓	HS↑
Fixed final	17.9702 ± 0.1733	29.7277 ± 0.3482	44.4931 ± 0.2535
Fixed multiple	17.8354 ± 0.0425	29.5402 ± 0.1076	44.9313 ± 0.0704
Free alignment	17.8239 ± 0.0499	29.5768 ± 0.1070	44.9562 ± 0.2465
Aggregate only	17.8615 ± 0.0681	29.6399 ± 0.0659	44.8466 ± 0.1760
Gate only	17.8308 ± 0.0477	29.5443 ± 0.0661	44.8715 ± 0.1831
Both constraints	17.8074 ± 0.0448	29.5550 ± 0.1039	44.9811 ± 0.1267

Panel B: diagnostics

Target family	Target dist.	Gate err.	Agg. err.
Fixed final	0.71394	0	0
Fixed multiple	0.68507	0	0
Free alignment	0.33302	14.005	0.44808
Aggregate only	0.60598	10.096	0.042863
Gate only	0.36291	1.6342e-13	0.44545
Both constraints	0.60699	1.0481e-13	1.1468e-14

Table 9 reports quality and diagnostics. Appendix F.1 reports all 12 endpoints and paired differences. Mean rankings alone do not establish significance.

F.1 SEED-LEVEL QUALITY AND REGISTERED CONTRASTS

Positive differences below indicate improvement: reference minus target for perplexity, target minus reference for accuracy. SD uses $n - 1$ with $n = 2$.

Table 10: Every retained E4 seed and quality metric; suffixes 910/911 denote 20260910/20260911.

Variant	Seed	WT2	C4	HS (%)
Fixed final exit	910	17.8476	29.4815	44.6724
Fixed final exit	911	18.0927	29.9740	44.3139
Fixed multiple exits	910	17.8655	29.4641	44.9811
Fixed multiple exits	911	17.8053	29.6163	44.8815
Free alignment	910	17.8592	29.5011	44.7819
Free alignment	911	17.7886	29.6525	45.1305
Aggregate only	910	17.9097	29.5933	44.7222
Aggregate only	911	17.8133	29.6865	44.9711
Gate only	910	17.8646	29.4976	44.7421
Gate only	911	17.7971	29.5911	45.0010
Both constraints	910	17.8391	29.4815	44.8915
Both constraints	911	17.7757	29.6284	45.0707

Table 11: Paired improvement: mean $\pm$ sample SD, with the number of seeds improving in parentheses. No composite metric or significance claim.

Comparison	WT2	C4	HS (pp)
fixed_multi vs. fixed_final	0.1348 ± 0.2159 (1/2)	0.1876 ± 0.2406 (2/2)	0.4382 ± 0.1831 (2/2)
free vs. fixed_multi	0.0115 ± 0.0074 (2/2)	-0.0366 ± 0.0005 (0/2)	0.0249 ± 0.3169 (1/2)
both vs. free	0.0165 ± 0.0052 (2/2)	0.0218 ± 0.0032 (2/2)	0.0249 ± 0.1197 (1/2)
both vs. gate	0.0234 ± 0.0030 (2/2)	-0.0106 ± 0.0378 (1/2)	0.1095 ± 0.0563 (2/2)
both vs. aggregate	0.0541 ± 0.0234 (2/2)	0.0849 ± 0.0379 (2/2)	0.1344 ± 0.0493 (2/2)

F.2 INTERPRETATION AND COST

Both constraints preserve gate scores and actual aggregate logits to numerical precision. Aggregate-only preserves the original weighted sum (residual below 2.02×10^{-14}), but changed gate masses yield actual logit errors up to 0.042863. Gate-only leaves logit error up to 0.44545. Free alignment lowers target distance more, while its mean C4 is worse than fixed-multiple targets. Both constraints have the best mean WT2 and HellaSwag; fixed-multiple targets have the best C4. The direction and size of the paired effects vary by metric and seed.

Measured costs include qualification runs, successful runs, failures and interrupted attempts: 36.0394 training process-hours and 6.2433 evaluation process-hours. These sums include overlapping jobs and exclude initializer search; they are not exclusive device-hours or complete recipe costs.

G TARGET CONSTRUCTION AT IDENTICAL STUDENT STATES

We compare targets before student calibration and after 256 parameter updates, using Ouro seed 20260910 and Parcae/LoopFormer seed 20260909. Each snapshot reuses identical teacher/student states from 32 fixed calibration windows at native depth. Sampling stride is 32: 64 positions on Ouro/Parcae and 32 on LoopFormer. This diagnostic sampling is distinct from Parcae's training stride 16. Each target-construction rule is applied at both student snapshots. No weights change between target variants.

Ouro compares fixed-final, fixed-multiple, unconstrained, weighted-sum-only, gate-only and both-constraint targets. The two fixed variants produce identical target states but use different training-loss normalizers; here all variants use the same all-state teacher-energy denominator. Final-state

models compare fixed-multiple, free and final-state-constrained targets. This yields 768 window-level comparisons across target families and snapshots. Figure 4 shows target-error reduction and teacher-output change for original targets, unconstrained alignment and Megrez. Each percentage is computed within a window and then averaged; output change is a relative logit L_2 norm, not the fraction of changed predictions. The snapshots are shared across target variants, rather than obtained by training each variant separately. Table 12 retains the single-constraint controls.

Table 12: Ouro-1.4B constraint controls at the two student snapshots in Figure 4. Both metrics are percentages averaged over 32 windows: target-error reduction and relative native-FP16 logit L_2 change. Before/after refer to student calibration, with 256 parameter updates. The fixed-final and fixed-multiple targets coincide here and share the original-target row.

	Before calibration		After calibration	
Target construction	Reduction ↑	Output change ↓	Reduction ↑	Output change ↓
Original teacher targets	0.0	0	0.0	0
Unconstrained alignment	84.6	45.01	76.6	38.64
Weighted-sum constraint only	21.8	3.04	23.3	2.74
Gate constraint only	79.7	44.75	71.4	38.80
Megrez (both constraints)	20.2	0.048	21.6	0.048

Table 13: Readout-constrained target reconstruction before student calibration and after 256 parameter updates. Errors use the common all-state teacher-energy denominator. Values are means over 32 windows. Time covers CPU target construction with two BLAS threads, excluding trajectory collection, data transfer and native-readout checks.

Model	Snapshot	Remaining error	Fixed component	Removed error	ms/window
Ouro-1.4B	Before	0.38772	0.14167	0.09808	104.6
Ouro-1.4B	After	0.36558	0.15513	0.10042	104.0
Parcae-1.3B	Before	0.04304	0.00577	0.01997	249.8
Parcae-1.3B	After	0.02746	0.00434	0.01370	249.7
LoopFormer 3×8	Before	0.03877	0.00452	0.03460	166.8
LoopFormer 3×8	After	0.01948	0.00120	0.02420	164.4

The remaining error decomposes into the fixed-subspace residual and the unremoved residual in its orthogonal complement (Eq. (2)). Across these snapshots, Megrez reduces squared target error by 20.2–55.5% on average per window. The FP64 solver preserves the specified reader to numerical precision; native FP16 Ouro evaluation has small rounding drift. For Parcae, the raw terminal teacher state is copied back after the roundoff check and substituted into the full native coda sequence, with other context and random state unchanged. This preserves the anchor exactly in native BF16.

H Schedule Selection and Fresh Confirmation

H.1 Development rule and document populations

We test two changes to Ouro's original schedule: refresh targets every update, or keep $\lambda = 0.03$ constant. Initializer, shared-weight optimizer, calibration windows and the update-256 endpoint are unchanged. Development uses seed 20260925, with KL-only and original Megrez as controls.

We sample 320 eligible documents from the first English C4 training shard (seed 20260925), each at least 2048 tokens under all three tokenizers, and freeze a 64/256 development/confirmation split before evaluation. Models share document identities and crop fractions, with tokenizer-specific positions. Documents and complete token windows that exactly match the available calibration or evaluation data, including C4 validation documents, are excluded. This is exact-match exclusion over recorded sources, not fuzzy deduplication or a claim of pretraining-data separation.

We select the candidate with the lowest development mean NLL among those that improve on both original Megrez and KL-only; if none qualifies, we retain the original schedule. The mean development NLLs (nats per predicted token) are 3.391387 for KL-only, 3.391505 for original Megrez,

3.393812 for every-step refresh and 3.393783 for constant $\lambda = 0.03$. Neither candidate qualifies, so the original schedule is retained.

Selection is frozen before confirmation and secondary benchmarks. The two controls train on A100 PCIe and the candidates on A100 SXM, so the screen does not isolate schedule from execution context. Selected and original Megrez coincide and are run once per confirmation seed.

H.2 Matched endpoints and statistical unit

Ouro and Parcae use fresh seeds 20260926/27, comparing KL-only, fixed-multiple states, free alignment and selected Megrez. Initializer, data order, optimization and numerical policy match within each model/seed; Parcae transfers the schedule without another search. Ouro's two seeds train on A100 PCIe/SXM, respectively, with all objectives sharing the device within a seed. Parcae uses the scale-Adam/FP64 assignment safeguards for every arm. LoopFormer evaluates its existing three KL-only/Megrez pairs on the new documents without retraining or another initializer choice.

Each document contributes one mean next-token loss. Ouro/Parcae use one 2048-token crop (2047 predictions); LoopFormer uses two consecutive 1024-token chunks (2046 predictions), kept together as one document. Inputs, precision, random-state policy and evaluation device match within each pair. We average paired seed effects within each document and bootstrap the 256 documents 10,000 times (seed 20260925; linear 2.5/97.5 percentile quantiles). These pointwise intervals condition on the trained endpoints and are unadjusted for multiple comparisons. Seed differences describe a separate source of variation; seeds are not treated as documents.

Table 14: Fresh-document NLL contrasts on 256 confirmation documents. Differences are first minus second (lower is better). Selected Megrez equals the original schedule. S26/S27 denote seeds 20260926/27. Intervals use 10,000 paired document resamples and condition on the trained models; they are pointwise and unadjusted. All registered two-seed contrasts are shown.

Model	Contrast	ΔNLL	95% document interval	S26	S27
Ouro-1.4B	Selected / KL-only	+0.00646	[+0.00576, +0.00728]	+0.00083	+0.01210
	Selected / Fixed	+0.00466	[+0.00394, +0.00543]	-0.00190	+0.01121
	Selected / Free	+0.00740	[+0.00674, +0.00813]	+0.00128	+0.01351
	Free / Fixed	-0.00274	[-0.00312, -0.00239]	-0.00318	-0.00230
Parcae-1.3B	Selected / KL-only	+0.01898	[+0.01086, +0.02641]	+0.05826	-0.02030
	Selected / Fixed	+0.02561	[+0.01876, +0.03197]	+0.04148	+0.00974
	Selected / Free	+0.03529	[+0.02910, +0.04119]	+0.05876	+0.01183
	Free / Fixed	-0.00968	[-0.01514, -0.00471]	-0.01727	-0.00209

Table 15: Original LoopFormer cohort evaluated on the 256 confirmation documents without retraining. Differences are Megrez minus KL-only. S09–S11 denote seeds 20260909–11. The pointwise, unadjusted document interval conditions on these three trained pairs.

Contrast	ΔNLL	95% document interval	S09	S10	S11
Original / KL-only	+0.00033	[+0.00011, +0.00054]	+0.00039	+0.00026	+0.00033

Free alignment has the lowest mean document NLL on Ouro and Parcae. On Parcae, fixed states and free alignment outperform Megrez in both seeds, while its difference from KL-only changes sign.

H.3 LoopFormer matched target-family extension

To complete the fixed/free comparisons, we run a separate four-objective cohort: KL-only, Megrez, fixed teacher states and free alignment, each with seeds 20260909–11 and 256 updates. All arms share the native TWLA-T initializer, calibration windows, within-seed data order, BF16 forward and FP32 master weights. The parent Lamb optimizer, value/assignment learning rates $3 \times 10^{-4}/3 \times 10^{-3}$, cosine schedule, refresh interval of eight and linearly decaying auxiliary coefficient (0.03 to zero) are retained. Fixed targets preserve all eight teacher states; the loss normalization and cache path match Megrez and Free. Only the final update enters the document comparison; development scoring every 32 updates does not select an endpoint or schedule.

All four arms within a seed train on the same device: H100 for seed 20260909, A100 PCIe for 20260910 and A100 SXM for 20260911. Thus variation across these pairs includes both seed and training-device effects. All 12 endpoints are evaluated on H100 using the same 256 C4 documents and two-chunk loss definition above. This is a post hoc extension on an already reported document population, not independent unseen confirmation. Tables 16 and 17 retain every registered contrast, seed difference and endpoint mean. Historical LoopFormer cohorts, including the $+0.000326$ frozen-pair difference above, remain separate.

Table 16: LoopFormer matched reruns: all four registered NLL contrasts after 256 updates on the already reported 256-document C4 population. Differences are first minus second; positive values favor the second. Intervals use 10,000 paired document resamples conditional on these endpoints. S09–S11 denote seeds 20260909–11; SD measures variation across these three seed/device pairs, separately from document uncertainty.

Contrast	ΔNLL	95% doc. interval	S09	S10	S11	SD
Megrez − KL-only	-0.000241	[-0.000442, -0.000040]	-0.000153	-0.000229	-0.000340	0.000094
Megrez − Fixed	-0.000014	[-0.000191, +0.000162]	+0.000120	+0.000253	-0.000414	0.000353
Megrez − Free	-0.000403	[-0.000596, -0.000214]	+0.000217	-0.000337	-0.001089	0.000656
Free − Fixed	+0.000389	[+0.000199, +0.000583]	-0.000097	+0.000590	+0.000675	0.000423

Table 17: Every arm in the new LoopFormer rerun cohort on the already used document population. NLL is in nats per predicted token. Means average the three seeds; SD covers these seed/device pairs. Historical endpoints remain a separate cohort.

Objective	Mean NLL	S09	S10	S11	SD
KL-only	3.653612	3.653476	3.653969	3.653392	0.000312
Megrez	3.653371	3.653322	3.653740	3.653051	0.000347
Fixed	3.653385	3.653202	3.653488	3.653465	0.000159
Free	3.653774	3.653105	3.654078	3.654140	0.000580

The new mean Megrez-minus-control NLL differences are -0.000241 versus KL-only, -0.000014 versus Fixed and -0.000403 versus Free. All three KL-only pairs favor Megrez, while Fixed and Free comparisons change sign across pairs. The document interval for the Fixed comparison includes zero; the other two exclude zero conditional on these checkpoints. The 12 training runs take 487.9–935.7 seconds each with 4421–4468 MiB peak allocated memory, with four jobs sharing each device; document evaluation takes 46.2–47.2 seconds per endpoint with 808 MiB peak reserved memory.

H.4 BENCHMARK CONFIRMATION

Table 18: Every new-seed confirmation endpoint. WT2/C4 are perplexities; HS is normalized accuracy in percent. Seed suffixes 26/27 denote 20260926/27. All arms use 256 updates and the same initializer within each model. Historical primary cohorts remain separate.

Model	Objective	Seed	WT2	C4	HS
Ouro-1.4B	KL-only	26	17.7771	29.7159	45.1205
	KL-only	27	17.8315	29.5579	44.9313
	Megrez	26	17.7687	29.7305	45.1703
	Megrez	27	18.0375	29.9154	44.5728
	Fixed states	26	17.8110	29.7858	45.0309
	Fixed states	27	17.8570	29.5894	44.9512
	Free alignment	26	17.7873	29.6757	44.7421
	Free alignment	27	17.8314	29.5299	44.9114
Parcae-1.3B	KL-only	26	20.1281	28.3513	44.3637
	KL-only	27	20.4082	29.0147	43.5969
	Megrez	26	21.1016	29.9548	43.4674
	Megrez	27	20.1340	28.5032	44.0351
	Fixed states	26	20.5314	28.9369	43.9554
	Fixed states	27	19.9991	28.3904	44.2243
	Free alignment	26	20.9261	28.6257	44.1247
	Free alignment	27	19.8441	28.1306	44.2840

Table 19: Remaining metrics for the same four Megrez endpoints. Together with Table 18, these cover all ten registered metrics. LamP is perplexity; other columns are accuracy in percent. Component controls were registered only for WT2/C4/HS.

Model	Seed	LamP	ARC-C	ARC-E	PIQA	WG	Lam	MMLU
Ouro-1.4B	26	37.3911	30.0341	52.9040	63.0033	52.1705	35.4163	25.6801
Ouro-1.4B	27	38.1296	28.7543	53.9983	63.3841	50.5919	35.5909	25.0534
Parcae-1.3B	26	50.8403	27.9863	54.5875	65.8324	53.7490	28.5853	24.0564
Parcae-1.3B	27	49.6632	29.0956	54.7138	66.1045	52.7230	28.3330	24.2202

The controls cover WT2/C4/HellaSwag, and Megrez covers all ten metrics. Perplexities are averaged after exponentiation per seed; accuracies and populations follow Appendix A. Ouro's lowest mean WT2 comes from KL-only and lowest C4 from free alignment. On Parcae, fixed states and free alignment outperform Megrez on all three component metrics in each new seed. Fixed states have the lowest mean WT2, while free alignment has the lowest C4 and highest HellaSwag. Secondary benchmark evaluations of the development endpoints occur after selection and are not pooled into confirmation means.

H.5 OPTIMIZATION DIAGNOSTICS AND MEASURED COSTS

At updates 1, 8, 64, 128, 192 and 255, we measure KL and auxiliary gradients and target motion. The scaled norm ratio is $\lambda\|g_{\mathrm{aux}}\|/\|g_{\mathrm{KL}}\|$ before optimizer preconditioning. Cached-to-fresh target motion uses the original teacher norm on Ouro and the cached target norm on Parcae (including FP32 cache rounding).

In the original development schedule the scaled gradient ratio is at most 1.86% over the six measured updates. Every-step refresh removes the measured target staleness, while a constant coefficient increases late auxiliary influence; neither meets the development selection rule. For detached targets and residual vector $\mathbf{e}$, the chain rule gives $g_{\mathrm{aux}} = 2J_\theta^\top \mathbf{e}/c$, with c the implemented target-energy normalizer. The Jacobian maps state-space residuals to parameter gradients. At Parcae's first update, seed 20260926, free alignment reduces target error from 0.03898 to 0.00585 while the unscaled auxiliary gradient norm rises from 7.888 to 8.782; the two KL gradients coincide.

Per-run time and peak reserved memory are 2.799–3.012 h and 31.680–32.242 GiB for the eight Ouro confirmation runs, 0.327–0.370 h and 12.340–12.432 GiB for the eight Parcae runs, and 2.418–3.548 h and 31.441–32.242 GiB for the four Ouro development runs. Times include loading, calibration, diagnostics, export and development scoring, excluding initializer production and

independent evaluation. Ouro shares A100 cards (up to two jobs); Parcae uses one job per RTX 4090 D. These differing execution conditions prevent a controlled speed comparison.

H.6 Hardware reproduction and frozen-weight evaluation

Table 20: Supplementary hardware reproduction, training seed 20260926 and 256 updates. The A100 rows are the corresponding primary-study endpoints, not additional runs. All four endpoints are evaluated on the same H100 with the same native precision and inputs. NLL averages 256 fresh documents; WT2/C4 are perplexities and HS is accuracy in percent. The two device contexts remain separate; one training seed cannot quantify seed variability.

Training GPU	Objective	Document NLL	WT2	C4	HS
A100 PCIe	KL-only	3.276316	17.7771	29.7159	45.1205
A100 PCIe	Megrez	3.277141	17.7687	29.7305	45.1703
H100	KL-only	3.277690	17.7745	29.7700	45.1006
H100	Megrez	3.276101	17.7706	29.6942	44.9811

A separate Ouro H100 training pair repeats seed 20260926, the initializer, data order and 256-update schedule. Evaluating both training pairs on H100 gives Megrez-minus-KL NLL -0.00159 for the H100-trained pair and $+0.00083$ for the A100-trained pair. HellaSwag favors KL-only for the H100-trained pair. This sign change is observed in one seed per context and does not isolate a causal GPU effect.

Frozen-weight evaluation. All eight new Ouro checkpoints are additionally evaluated on RTX 4090 D, with identical weights, FP16 native readout, task populations and batch sizes. Across endpoints, maximum absolute H100/RTX differences are 0.00029/0.00115 WT2/C4 perplexity and 0.07967 HellaSwag percentage points. These repeats test evaluation reproducibility, not training invariance; primary H100 evaluations are retained without pooling device contexts.

I Supplementary Scale Check: Ouro-2.6B

Ouro-2.6B has 48 physical blocks and 336 shared projections, versus 24 and 168 in Ouro-1.4B. Hidden width 2048, four calls and the gated reader remain unchanged; the target solver's active subspace is still at most eight-dimensional.

I.1 Identical-state target comparison

We use two fixed student models: a historical 256-update Megrez continuation (seed 20260909, affine RTN initialization) and a single-plane TWLA-T initializer. They are distinct student snapshots, not matched training interventions. All six target families reuse the first 32 calibration windows of 2048 tokens, with 64 positions at stride 32. The common error denominator is the teacher's squared pre-gain norm across all four states.

Table 21: Target construction on Ouro-2.6B. Each target family receives the same frozen states in each of 32 calibration windows. Error reduction is the mean per-window reduction in squared all-state mismatch. Ideal drift is the maximum relative logit error in FP64; native drift is the mean after the FP16 readout. Fixed-final and fixed-multiple targets coincide in this diagnostic, so one fixed row represents both.

Target family	Error reduction (%)	Remaining error	Ideal drift	Native drift
RTN-start continuation				
Fixed targets	0.00	1.54262	0	0
Free alignment	93.63	0.09821	8.96×10^{-1}	7.60×10^{-1}
Aggregate only	25.39	1.15109	4.49×10^{-1}	3.94×10^{-1}
Gate only	79.94	0.30954	1.10×10^{0}	9.55×10^{-1}
Both constraints	17.42	1.27404	1.06×10^{-14}	4.56×10^{-4}
TWLA-T initialization				
Fixed targets	0.00	0.64308	0	0
Free alignment	95.27	0.03040	5.57×10^{-1}	5.30×10^{-1}
Aggregate only	18.53	0.52412	3.98×10^{-2}	3.15×10^{-2}
Gate only	87.57	0.07997	5.34×10^{-1}	5.05×10^{-1}
Both constraints	16.27	0.53868	1.21×10^{-14}	4.58×10^{-4}

Both constraints remove 17.42%/16.27% of mean per-window mismatch at the continuation/TWLA snapshots. Maximum ideal logit drift is at most 1.22×10^{-14}; native FP16 mean drift is about 4.6×10^{-4}. Free alignment reduces distance further while changing the teacher's output. Both-constraint CPU construction takes about 106 ms/window, excluding state collection and under co-resident work.

I.2 Prediction and device repeats

All evaluations use the native four-call FP16 readout, batch one, and fixed 2048-token windows (148 WT2, 256 C4). Cross-entropy is FP32 with FP64 summation. Device repeats use identical weights, not extra training seeds.

Table 22: Native Ouro-2.6B evaluation on frozen endpoints. All entries use four recurrent calls, FP16 model execution and FP32 cross-entropy with FP64 accumulation. WT2 has 148 windows and C4 has 256, each of length 2048. The continuation uses its historical 256-update, single-seed endpoint. The A100 rows repeat two of the same frozen models.

Endpoint	Device	WT2 PPL↓	C4 PPL↓
FP16 teacher	RTX4090D	7.5904	11.3891
TWLA-T initializer	RTX4090D	77.7184	141.4098
Megrez (RTN start)	RTX4090D	301.5178	1074.6358
FP16 teacher	A100 PCIe	7.5903	11.3890
Megrez (RTN start)	A100 SXM	301.5206	1074.6490

The TWLA-T initializer has lower perplexity than the historical continuation. The continuation starts from affine RTN, so this comparison does not isolate the calibration objective under matched initialization.

J Additional Mechanistic Comparisons

Geometry of supervision. Output distillation matches teacher predictions (Hinton et al., 2015); FitNets and Patient KD supervise intermediate or normalized features (Romero et al., 2015; Sun et al., 2019). Relational KD matches pairwise distances and triplet angles (Park et al., 2019), while similarity-preserving KD matches minibatch Gram matrices (Tung & Mori, 2019). Megrez preserves geometry across one token's recurrent states and constrains their native readout, rather than matching relations across examples.

Other calibration and execution changes. GPTQ and OmniQuant optimize rounding or learnable transformations (Frantar et al., 2023; Shao et al., 2024). AYOT changes calibration examples and generated context (Wang et al., 2026a); FlashLoop changes recurrent execution and KV storage (Yang & Liu, 2026).