LEARNING DYNAMICS OF LLM CALIBRATION UNDER SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Calibrated confidence helps assess the reliability of an LLM's response. In the literature, LLM fine-tuning is often associated with poor calibration, but previous analyses only compared the calibration performance of pre-trained and post-trained models, leaving the training dynamics largely unexplored. In this work, we study the calibration dynamics of LLMs during standard supervised fine-tuning (SFT), and find that the calibration of an overconfident LLM is improved first and then worsens during training. The underlying reason is that hard training questions that LLMs generate incorrect responses can improve calibration, but their influence weakens as their proportion decreases during training. Based on this insight, we propose State-Gated Corrective Replay (SGCR), a novel method that amplifies the impact of those hard training questions through oversampling. In effect, SGCR can reduce calibration error and improve calibration stability during training, thereby extending the iteration window with excellent calibration. Extensive experiments on MMLU and MedMCQA validate the effectiveness of our method, reducing time-averaged ECE by 36.0–47.6% relative to standard SFT across the evaluated models.

1 INTRODUCTION

Confidence calibration matters when users need to judge whether to trust an answer from a large language model (LLM). A calibrated model assigns confidence scores that match how often its answers are correct (Guo et al., 2017; Jiang et al., 2021). This relationship needs to be considered when models are trained to follow instructions or solve specific tasks. Supervised fine-tuning (SFT), a standard method for adapting models to these tasks, increases the probability of reference answers and therefore changes the model's confidence as well as its predictions (Wei et al., 2022b; Ouyang et al., 2022). To understand whether SFT makes a model more reliable, we need to consider how it changes calibration alongside task accuracy.

Fine-tuning is often viewed as a cause of poor calibration. The GPT-4 Technical Report showed poorer calibration after reinforcement learning from human feedback (RLHF; OpenAI, 2023). Similar concerns have been reported for instruction-tuned and other post-trained models, which can assign high confidence to incorrect answers (Huang et al., 2025; Luo et al., 2025). Prior work improves calibration by adding calibration regularization to the training loss (Xiao et al., 2025) or applying label smoothing (Huang et al., 2025). By varying the proportions of data classified as known or unknown before fine-tuning, CogCalib finds that training on known data can induce overconfidence, whereas training on unknown data can improve calibration (Wang et al., 2025). However, SFT's effect on calibration can change during training even with the same dataset and standard negative log-likelihood (NLL) loss. We find that when a model is overconfident before training, calibration improves early and worsens later, while the training data and NLL loss remain unchanged (Figure 1). This raises a question: why does the early improvement fade, and how can it be sustained?

We find that updates from training questions the model gets wrong improve calibration on held-out questions on average. Updates from questions it gets right tend to do the opposite by increasing confidence, including confidence in wrong answers on the evaluation set. As the model learns, it gets more training questions right, so fewer questions it still gets wrong appear in randomly sampled batches. The norm of the mean gradient from these questions is also much smaller late in training than at the start. These gradients still favor better calibration, but they have less influence on the

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.

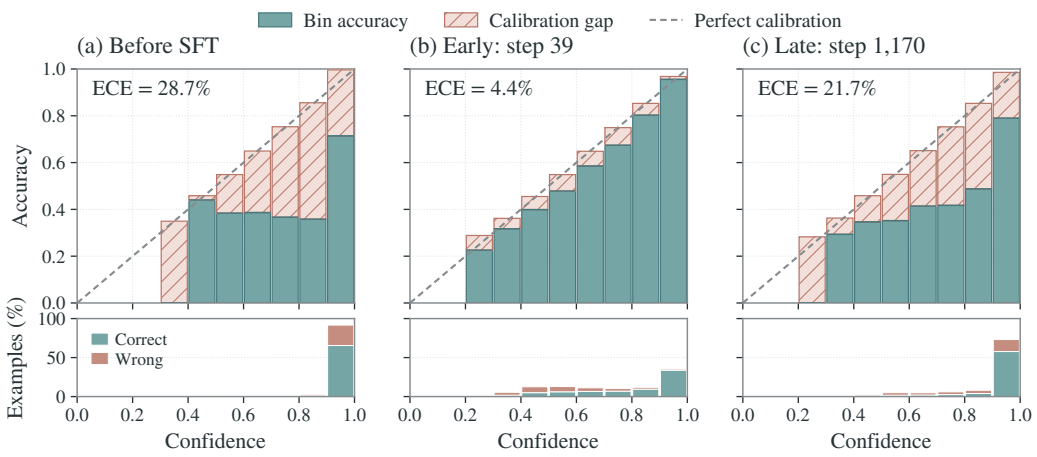


Figure 1: **Calibration improves early and worsens later.** Qwen3-4B Instruct on the MMLU test set, seed 1. Top: bin accuracy, with hatching for the gap to mean confidence. Bottom: the fraction of questions in each bin, split by correctness. All panels use ten equal-width bins; empty bins are omitted. Appendix F shows results for other seeds and Base.

overall update. This helps explain why the early improvement fades while updates from correctly answered questions continue to raise confidence. We also find that rescaling logits before training can reverse SFT's effect on calibration, even when the model's initial predictions barely change. The scale used in the loss changes the training gradients, whereas scaling only at evaluation changes measured confidence.

To preserve the early calibration gains, we propose State-Gated Corrective Replay (SGCR), which reserves part of each training batch for questions the model still gets wrong. It checks these questions again before replay and leaves the NLL loss unchanged. SGCR aims to reduce calibration error and extend the calibration window, a sequence of consecutive evaluated checkpoints below a chosen ECE threshold.

Experiments on MMLU and MedMCQA demonstrate that SGCR improves calibration over standard SFT and extends the calibration window. For example, it reduces Llama-3.2-3B's time-averaged ECE on MMLU from 13.55% to 7.10%. Similarly, it lowers Qwen3-4B's time-averaged ECE on MedMCQA from 7.36% to 4.09%. For a given training dataset, extending this window makes more checkpoints available that satisfy a chosen calibration requirement.

Our main contributions are as follows:

1. We show that standard SFT can improve calibration early and worsen it later when a model is too confident before training. Both phases occur with the same training dataset and NLL loss, revealing an early improvement that comparisons between the starting and final models can miss.
2. We find that updates from training questions the model currently gets wrong tend to improve calibration on held-out data, while updates from questions it currently gets right tend to worsen it. As training proceeds, the questions it still gets wrong become less common and have less influence on the overall update. We also show that rescaling logits before training can reverse the effect of SFT on calibration.
3. We propose SGCR, which increases how often the model trains on questions it still gets wrong while retaining the standard NLL loss. We show that this sampling intervention can reduce calibration error and extend the calibration window during continued training.

2 Preliminary

For a multiple-choice question x, let $\ell_\theta(x) \in \mathbb{R}^V$ be the full-vocabulary logits at the answer-token position. The selector S_x retains the K legal answer logits, and $H = I_K - \mathbf{1}\mathbf{1}^\top/K$ removes their

common offset. Define candidate probabilities, prediction, confidence, and correctness by

$$z_x = HS_x\ell_\theta(x), \quad q_x = \mathrm{softmax}(z_x), \quad k_x = \arg\max_a q_{x,a}, \quad c_x = q_{x,k_x}, \quad a_x = \mathbf{1}[k_x = y_x]. \tag{1}$$

Here y_x is the gold candidate index; centering preserves probabilities and answer ranking.

A calibrated model satisfies $\Pr(a_x = 1 \mid c_x) = c_x$ (Guo et al., 2017; Jiang et al., 2021). On n evaluation questions, we use ten equal-width confidence bins I_b, with $n_b = |I_b|$, and estimate expected calibration error as

$$\mathrm{ECE} = \mathrm{ECE}_{10} = \sum_{b:n_b>0} \frac{n_b}{n} \left| \underbrace{\frac{1}{n_b}\sum_{x\in I_b} a_x}_{\mathrm{acc}(I_b)} - \underbrace{\frac{1}{n_b}\sum_{x\in I_b} c_x}_{\mathrm{conf}(I_b)} \right|. \tag{2}$$

Bins are left-closed and right-open except the last, $[0.9, 1]$; empty bins contribute zero. We also report accuracy, multiclass Brier score $n^{-1}\sum_x \sum_a (q_{x,a} - \mathbf{1}[a = y_x])^2$, and candidate NLL $-n^{-1}\sum_x \log q_{x,y_x}$. Lower Brier and NLL indicate better probability predictions.

Standard SFT in our answer-token setup minimizes

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{m}\sum_{i=1}^{m} \log\left[\mathrm{softmax}\big(\ell_\theta(x_i)\big)\right]_{y_i}. \tag{3}$$

Here y_i indexes the gold token in the full vocabulary. Training probabilities normalize over all V tokens; evaluation confidence normalizes over candidate answers.

The main experiments follow full-parameter SFT of Qwen3-4B Instruct and Base (Yang et al., 2025) on MMLU (Hendrycks et al., 2021), using separate training, validation, and test splits. We use AdamW with a cosine learning-rate schedule and evaluate throughout training. Instruct trajectories use three seeds; Base is a one-seed control. Replications use Llama-3.2-3B-Instruct and MedMCQA (Meta AI, 2024; Pal et al., 2022). Trajectory bands show variation across seeds, while intervention intervals show variation across batches at a fixed model and optimizer state. Appendix E gives dataset sizes and training settings; Appendix O retains earlier Qwen2.5 experiments.

3 Calibration Dynamics during SFT

Standard SFT can improve calibration before continued training reverses the gain. Figure 1 illustrates this change at three checkpoints, while Figure 2 follows ECE, Brier score, and candidate NLL throughout training for Qwen Instruct and Base on the same MMLU test set.

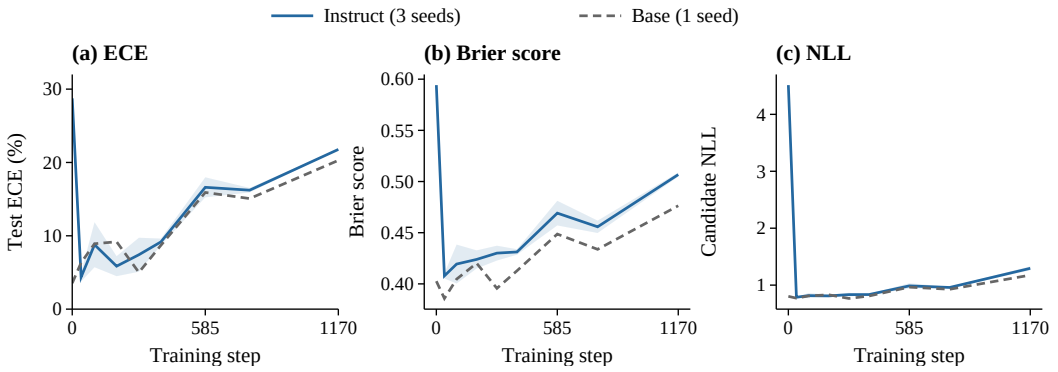


Figure 2: **Early gains in probability quality can fade as SFT continues.** Complete Qwen trajectories on the MMLU test set: (a) ECE-10, (b) Brier score, and (c) candidate NLL. Blue: Instruct three-seed mean ± SD. Dashed gray: one-seed Base control. Lower is better in all panels. NLL uses log-softmax without a probability floor.

Early improvement and later worsening. Figure 1 follows one Instruct run before SFT, early in training, and late in training. In the top row, the gap between mean confidence and bin accuracy narrows early and widens again later. The bottom row shows predictions spreading across confidence bins early, then concentrating in the highest-confidence bin again, including incorrect answers. Thus, the later model assigns high confidence to many errors despite the earlier improvement. Figure 2 confirms that this fall and rise in ECE occurs across all three Instruct seeds. Both phases use the same training dataset and NLL loss, without an added calibration objective.

Different model settings. Base is better calibrated at initialization but becomes less calibrated during training. Thus, the early gain is not a general consequence of applying SFT to any model: it depends on the initial probability predictions. This comparison motivates examining the starting logit scale in Section 4.3; the Instruct–Base contrast by itself does not isolate scale from other differences between the models.

Changes beyond ECE. Brier score and candidate NLL also improve early in the Instruct runs and worsen later. Their agreement with ECE indicates a broader change in probability quality, although these proper scores also depend on predictive performance and are not pure calibration measures. Base follows a different early path, with its proper scores also worsening later. Individual-seed curves, accuracy trajectories, and further replications are in Appendix G. The loss stays the same as training progresses, but the model answers more training questions correctly. We next examine how updates from questions the model gets right and wrong contribute to the change in held-out calibration.

4 Why Early Calibration Gains Fade

4.1 Updates from Questions the Model Gets Wrong

Updates from training questions the model gets wrong improve held-out calibration on average, while updates from questions it gets right tend to worsen it (Figure 3). We call these training groups Wrong (W) and Correct (C). Membership is determined by the model's candidate-answer prediction immediately before the update.

We isolate the effect of each group by restoring the same model parameters and AdamW state before every trial. From checkpoints spanning early and late training, we take one NLL update using only Wrong questions, only Correct questions, or a naturally sampled batch. Every condition uses the same batch size and validation questions. Repeating the comparison across batches and seeds separates the average effect of a group from variation in which questions are sampled.

In panel (b), Wrong-only updates lower the original wrong-answer probability and centered-logit norm and raise entropy on held-out questions; Correct-only updates have the opposite mean effects. We track the original answer k_t via $\Delta p_{\text{old}} = q_{t+1,k_t} - q_{t,k_t}$ even if the prediction changes, so this comparison follows the same answers through an update.

Training updates affect held-out predictions through shared parameters. With $p_i = \mathrm{softmax}(\ell_i)$ and $J_i = \partial \ell_i / \partial \theta$, the NLL residual is $\epsilon_i = p_i - e_{y_i}$ and $g_i = J_i^{\mathsf{T}} \epsilon_i$. Write $g = \sum_i w_i g_i$, where $w_i \geq 0$, $\sum_i w_i = 1$, and $J_x^c = \partial z_x / \partial \theta$.

Proposition 1 (Held-out logit changes after one update)**.** *Fix the parameters and an affine update direction $d = b + Pg$, with b, P fixed as g varies. If J_x^c is L_x-Lipschitz along the update segment, then*

$$\Delta z_x = -\eta J_x^c b - \eta \sum_i w_i K_P(x,i)\epsilon_i + e_x, \quad K_P(x,i) = J_x^c P J_i^{\mathsf{T}}, \quad \|e_x\| \leq \tfrac{1}{2} L_x \eta^2 \|d\|^2. \quad (4)$$

$K_P(x,i)$ maps a training residual to its first-order effect on held-out logits (Ren & Sutherland, 2025). SGD has $P = I, b = 0$; a frozen preconditioner gives an affine map. The general map in Section 5.1 accommodates AdamW's gradient-dependent denominator and gradient clipping. Proofs are in Appendix B.

These opposite effects also hold across Qwen and Llama on MMLU and MedMCQA (Table 1). MedMCQA uses official validation and a fixed selection rule favoring late checkpoints where natural updates increase ECE (Appendix L). Appendix K compares the same examples early and late.

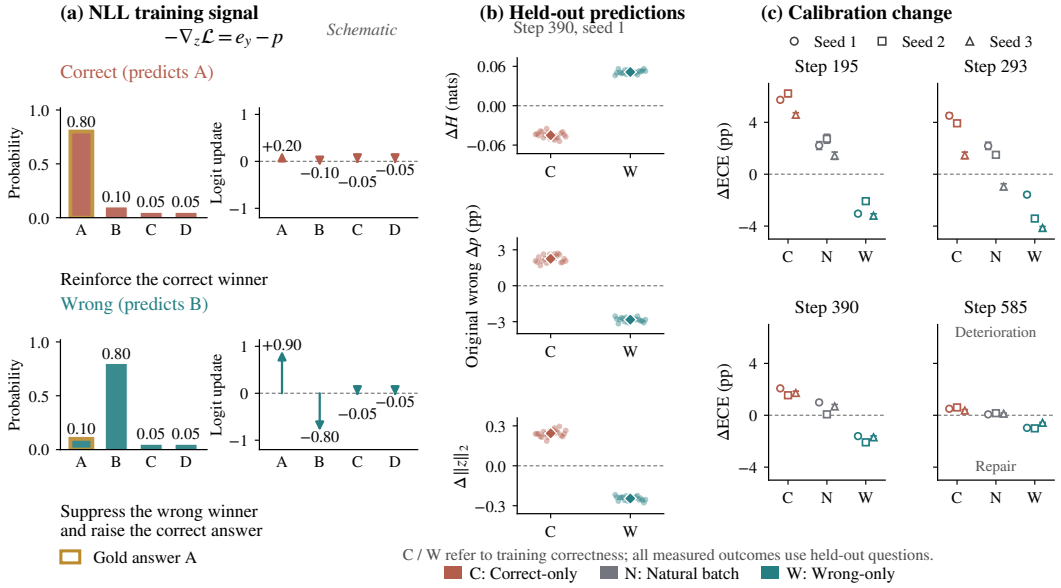


Figure 3: **Wrong-only updates reduce confidence in held-out wrong answers.** (a) NLL schematic, with gold answer A. Actual SFT uses full-vocabulary NLL. (b) Prediction changes at step 390, seed 1; small points are batches and diamonds are means. (c) ECE changes by batch type; shapes identify seeds. Intervals are pointwise 95% batch-bootstrap intervals over 20 trials. Appendix J.3 gives all prediction changes.

Table 1: **Calibration effects of group-only updates.** ECE-10 change in percentage points (negative values improve calibration): mean ± SD over three seeds, after averaging each seed's results across its tested checkpoints. Protocols differ, so effect magnitudes are not directly comparable.

Model	Training task	Wrong-only Δ ECE ↓	Correct-only Δ ECE ↓
Qwen3-4B	MMLU	-2.12 ± 0.31	$+2.77 \pm 0.65$
Llama-3.2-3B	MMLU	-3.79 ± 0.42	$+4.44 \pm 1.07$
Qwen3-4B	MedMCQA	-1.01 ± 1.16	$+4.14 \pm 0.57$
Llama-3.2-3B	MedMCQA	-3.17 ± 1.35	$+7.40 \pm 3.41$

4.2 Why These Updates Have Less Effect Later

The useful updates become rarer as the model learns. The fraction of training questions still answered incorrectly falls from about 12% early in training to below 1% at the end. In every seed, the norm of their mean parameter gradient ends far below its initial value. Figure 4 shows that this decline is not monotonic: the norm drops sharply at first and then rises later. The later rise occurs when far fewer wrong questions remain in naturally sampled batches.

To connect frequency, magnitude, and direction, let π_W be the current Wrong fraction and g_W, g_C the groups' mean NLL parameter gradients. Let $h = \nabla_\theta \mathcal{R}$, where $\mathcal{R}$ is a smooth validation calibration objective, defined as squared soft-bin reliability error in Appendix J. For nonzero gradients, the projection $\rho_G = \langle h, g_G \rangle / \|h\|$ is positive when an SGD group update lowers $\mathcal{R}$ to first order. Weighting by the group's frequency gives

$$\pi_W \rho_W = \pi_W \|g_W\| \cos(h, g_W). \tag{5}$$

We estimate group gradients on a fixed training pool and use the full training set to measure group proportions.

Wrong-group projections remain positive at every saved checkpoint and seed, whereas Correct-group estimates are negative. The direction of the Wrong gradient therefore continues to favor better calibration, but its influence in the naturally weighted update becomes much smaller. Fewer wrong questions, smaller gradients than at initialization, and weakening alignment act together. Combined with the restored-AdamW interventions, these measurements help explain how the early benefit can fade even while Wrong-only updates remain useful.

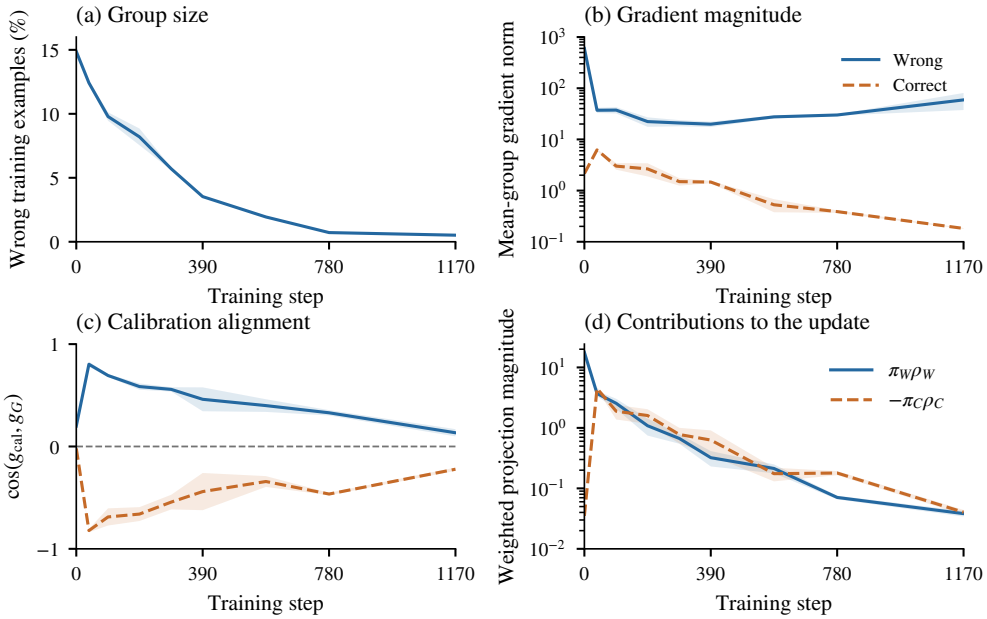


Figure 4: **Wrong gradients still favor calibration but have less influence later.** Qwen Instruct, three-seed mean ± SD. (a) Wrong fraction. (b) Group mean-gradient norms. (c) Alignment with the calibration gradient $g_{\text{cal}} = h$; positive values favor improvement under SGD. (d) Weighted projection magnitudes. Norms and magnitudes use logarithmic axes.

4.3 The Effect of Logit Scale

Changing the starting probabilities can reverse SFT's effect on calibration even when the predicted answers barely change. Scaling up Base makes its later calibration improve; scaling down Instruct greatly reduces its early improvement. We implement the change by rescaling the final RMSNorm weight using a validation-fitted temperature ratio. A positive scale preserves answer rankings in exact arithmetic, with a small discrepancy in the BF16 experiment. Appendix H gives the complete crossover curves and prediction agreement.

Table 2 separates the RMSNorm multiplier g from the loss-effective scale g/τ, where $\tau > 0$ is the loss temperature:

$$\mathcal{L}_{\tau,i} = \tau \,\mathrm{CE}(\ell_i/\tau, y_i), \qquad p_i = \mathrm{softmax}(\ell_i/\tau), \qquad \epsilon_i = \nabla_{\ell_i} \mathcal{L}_{\tau,i} = p_i - e_{y_i}. \tag{6}$$

The factor τ cancels the common $1/\tau$ gradient factor, and evaluation uses raw candidate logits. Table 2 shows similar accuracy but very different confidence when the output and loss-effective scales are mismatched. Matching the two scales gives much better calibration. Logit changes are also more similar between runs sharing the loss-effective scale than between runs sharing the output scale. Thus, the probabilities used in the loss affect how the model learns, while scaling only at evaluation changes the confidence reported for its predictions.

Table 2: **Mismatched scales yield very different confidence levels at similar accuracy.** Qwen Instruct after 390 updates, three-seed mean ± SD. High and low scales are 1 and $1/s$ ($s = 8.4629$). ↑/↓: higher/lower is better.

Output scale g	Loss-effective scale g/τ	Accuracy ↑ (%)	Confidence (%)	ECE-10 ↓ (%)	Brier ↓ score
High	High	70.2 ± 0.4	75.1 ± 0.7	4.9 ± 0.7	0.398 ± 0.004
High	Low	69.4 ± 0.3	96.2 ± 0.1	26.8 ± 0.4	0.560 ± 0.007
Low	High	70.2 ± 0.2	32.9 ± 0.2	37.2 ± 0.2	0.631 ± 0.003
Low	Low	69.5 ± 0.1	74.5 ± 1.0	5.0 ± 1.0	0.408 ± 0.003

To separate norm and direction, write $z_x = r_x u_x$ with $r_x = \|z_x\|_2 > 0$ and $\|u_x\|_2 = 1$. Increasing r_x with direction held fixed raises confidence (Wei et al., 2022a; Tian et al., 2021). For checkpoints s, t,

define $F_{ab}(x) = \max \mathrm{softmax}(r_a(x)u_b(x))$ and average over the two possible orders of changing norm and direction:

$$A_r(x) = \tfrac{1}{2}[(F_{ts} - F_{ss}) + (F_{tt} - F_{st})],$$
$$A_u(x) = \tfrac{1}{2}[(F_{st} - F_{ss}) + (F_{tt} - F_{ts})]. \tag{7}$$

Proposition 2 (Exact decomposition of the change in ECE)**.** *On a fixed evaluation set, define bin gaps $d_j = n^{-1}\sum_{x\in I_j}(c_x - a_x)$ and the total negative gap $U = \sum_j (-d_j)_+$. Then, even if predictions or bin memberships change,*

$$\mathrm{ECE} = \bar{c} - \bar{a} + 2U, \qquad \Delta\,\mathrm{ECE} = \overline{A_r} + \overline{A_u} - \Delta\bar{a} + 2\Delta U. \tag{8}$$

Applying this identity to the later training phase assigns most of the ECE increase to norm growth, with a smaller contribution from direction changes. Accuracy changes little, and the underconfidence term is negligible. Logit norm therefore explains much of the later confidence increase without requiring a large change in predicted answers. Appendix C gives the derivation and conditions; binned ECE need not equal population calibration error (Kumar et al., 2019).

Scale also changes which training questions contribute most to the output residual. Wrong questions dominate at high loss-effective scale, while Correct questions dominate at low scale. This connects the scale experiment to the group analysis above; the parameter update additionally depends on the Jacobians in Equation 4. Residual shares and optimizer controls are in Appendices H and I.

5 How to Extend the Calibration Window

5.1 Maintaining the Wrong Fraction as Training Progresses

As training progresses, maintaining the Wrong fraction requires adapting sampling to the model's changing answers. We first test how the Wrong fraction affects the next update by restoring the same model parameters and native AdamW state and varying only batch composition (Figure 5). Across all tested checkpoint–seed states, mean ECE change falls overall as the Wrong fraction increases, though not always monotonically. Naturally sampled batches usually worsen calibration at these states, but at least one higher Wrong fraction improves it at each state.

The intervention can also improve probability quality. At step 390, Wrong-only batches improve Brier score and NLL as well as ECE, with little average change in accuracy. Accuracy effects vary across checkpoints and become slightly negative on average later in training (Figure 19). Increasing the Wrong fraction directly controls the next calibration change, but its effect on task performance still depends on when the update is applied.

At a fixed checkpoint, write $g_\alpha = \alpha g_W + (1-\alpha)g_C$ for Wrong fraction α, with $h = \nabla_\theta \mathcal{R}$ as above and $A_G = h^\mathsf{T} P g_G$ for an affine optimizer.

Proposition 3 (A sufficient condition for one-step improvement)**.** *Let $d_\omega(g)$ be the optimizer direction with previous state ω fixed. If $\mathcal{R}$ has an L-Lipschitz gradient along the update segment, then*

$$\Delta\mathcal{R}(\alpha) = -\eta h^\mathsf{T} d_\omega(g_\alpha) + e_\alpha, \qquad |e_\alpha| \le \tfrac{1}{2}L\eta^2\|d_\omega(g_\alpha)\|^2. \tag{9}$$

Thus $h^\mathsf{T} d_\omega(g_\alpha) > \frac{1}{2}L\eta\|d_\omega(g_\alpha)\|^2$ ensures improvement. For $d = b + Pg$, the leading term is

$$-\eta\big[M + A_C + \alpha(A_W - A_C)\big], \qquad M = h^\mathsf{T} b. \tag{10}$$

If $A_W > A_C$, it decreases with α and crosses zero at $\alpha_* = -(M + A_C)/(A_W - A_C)$ when this lies in $[0, 1]$.

The bound retains the native optimizer map; experiments measure hard-binned ECE after AdamW. Proofs are in Appendix B; threshold comparisons from separately reconstructed runs are in Appendix J. Sustaining these useful updates requires repeatedly checking which questions remain wrong.

5.2 State-Gated Corrective Replay

State-Gated Corrective Replay (SGCR) maintains exposure to currently Wrong questions by adapting a small part of each training batch to the model's current answers. Algorithm 1 refreshes a pool

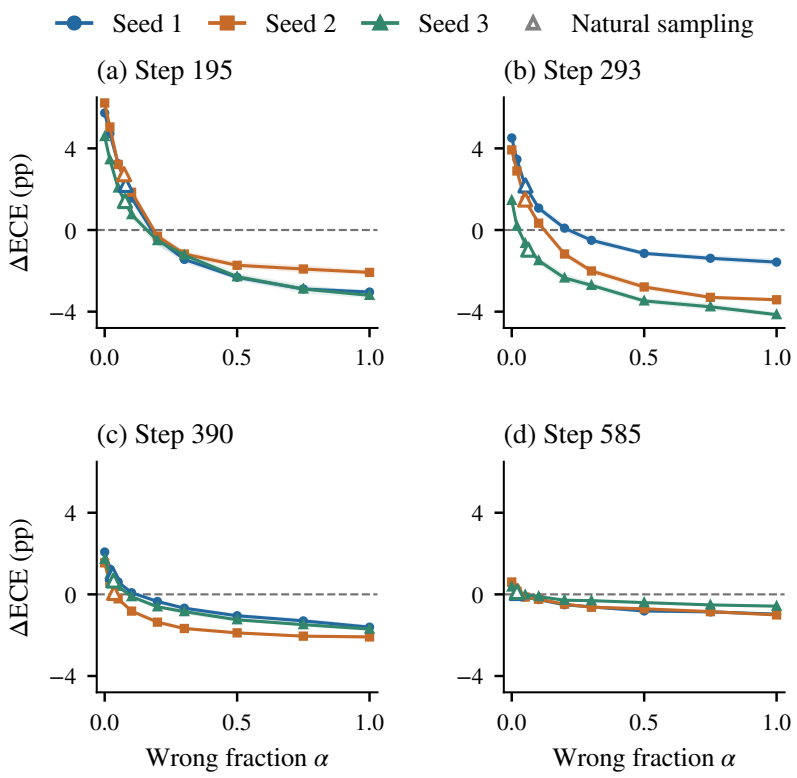


Figure 5: **Increasing the Wrong fraction can turn a harmful update into a beneficial one.** Each curve shows one-step validation ECE change over 20 batches from the same restored checkpoint and optimizer state. Bands are pointwise 95% batch-bootstrap intervals; triangles mark natural sampling. Appendix J gives gradient predictions and task metrics.

of current wrong questions, rechecks candidates before replay, and fills any shortfall uniformly. Questions answered correctly are excluded from replay but become eligible again if the model gets them wrong. This follows the broader idea of giving useful examples more training opportunities (Schaul et al., 2016), with eligibility tied to the model's current answer.

Algorithm 1 State-Gated Corrective Replay (SGCR)

Input: Training set $\mathcal{D}$, model θ, optimizer state ω, update budget T
Parameters: Batch size $B = 256$, replay quota $r = 26$, pool refresh interval 0.25 epoch
Output: Trained model θ

for $t = 0, \ldots, T-1$ **do**
  if $t = 0$ or the pool refresh is due **then**
    $\mathcal{W} \leftarrow \{i \in \mathcal{D} : k_\theta(i) \neq y_i\}$ ▷ scan the training set
  end if
  $\mathcal{U} \leftarrow \textsc{Sample}(\mathcal{D}, B - r)$; $\mathcal{S} \leftarrow \varnothing$; $Q \leftarrow \textsc{Shuffle}(\mathcal{W})$
  while $|\mathcal{S}| < r$ and Q is nonempty **do**
    $i \leftarrow \textsc{Pop}(Q)$
    if $i \notin \mathcal{U} \cup \mathcal{S}$ and $k_\theta^{\text{train}}(i) \neq y_i$ **then** ▷ recheck current answer
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$
    end if
  end while
  $\mathcal{F} \leftarrow \textsc{Sample}(\mathcal{D} \setminus (\mathcal{U} \cup \mathcal{S}), r - |\mathcal{S}|)$ ▷ uniform fallback
  $\mathcal{B} \leftarrow \mathcal{U} \cup \mathcal{S} \cup \mathcal{F}$
  Compute mean full-vocabulary NLL $\mathcal{L}_{\mathcal{B}}(\theta)$; audit $\mathcal{S}$ in this forward pass
  $(\theta, \omega) \leftarrow \textsc{AdamW}(\theta, \omega, \nabla_\theta \mathcal{L}_{\mathcal{B}}(\theta))$
end for
return θ

In Algorithm 1, SAMPLE draws uniformly without replacement, and k_θ^{train} is the training-context answer. At least 230 examples are sampled uniformly. The total Wrong fraction varies because these examples can also be Wrong and fallback can reduce replay below quota. NLL, gradient averaging, and AdamW stay unchanged. Scans and rechecks add forward passes; costs and audit results are in Appendix D.

5.3 EXTENDING THE CALIBRATION WINDOW

A *calibration window* is a consecutive sequence of evaluated checkpoints with ECE below threshold ε. On shared steps $t_0 < \cdots < t_m$, let $C_k = \text{ECE}_{10}(\theta_{t_k})$. The longest observed duration is

$$L_\varepsilon = \max_{0 \le i \le j \le m} \{t_j - t_i : \; C_k \le \varepsilon \text{ for all } k = i, \ldots, j\}, \tag{11}$$

or zero if no checkpoint qualifies. SGCR–Vanilla pairs share thresholds, evaluation steps, and horizons; crossing times are not interpolated. The illustrative 5% threshold is retrospective.

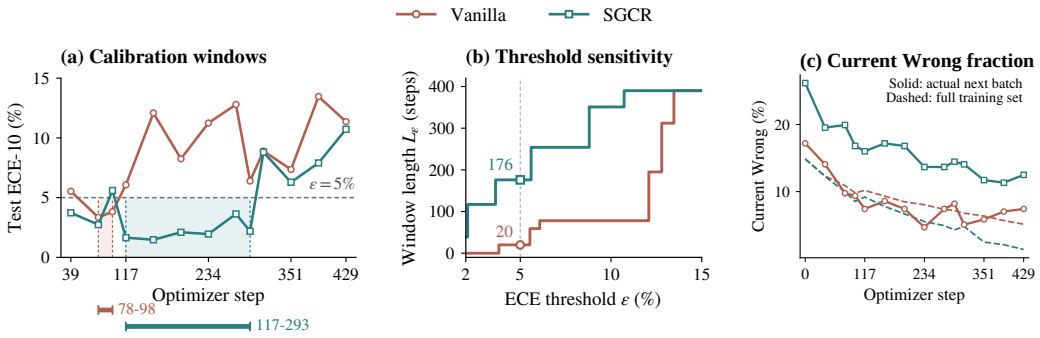


Figure 6: **SGCR extends the calibration window through corrective replay.** Qwen–MMLU, independent seed-1 pair through step 429. (a) Test ECE and 5% windows; initialization (28.85% ECE) is omitted. (b) Window length versus threshold. (c) Wrong fractions in the next training batch (solid) and the full training set at each checkpoint (dashed). Task metrics are in Appendix D.

SGCR maintains a higher Wrong fraction in training batches as the full-set fraction falls (Figure 6c). In Qwen–MMLU, time-averaged test ECE falls from 9.68% to 5.43%, while the longest window below 5% ECE grows from 20 to 176 updates. Appendix D defines the step-weighted average. Validation-set ECE yields the same window endpoints, with longer windows across the displayed thresholds. The shared horizon ends before SGCR finishes training. Lower accuracy and worse endpoint Brier and NLL remain tradeoffs.

Task tradeoffs vary across datasets. In the longer Qwen–MedMCQA comparison, SGCR extends the low-error window and improves endpoint ECE, Brier, and NLL with nearly unchanged accuracy. Llama also shows longer windows, though MedMCQA requires a looser threshold for a positive duration. These single-seed comparisons use official validation for MedMCQA; Appendices D.5 and D.6 report the trajectories, thresholds, and task metrics.

A longer window provides more checkpoints below a chosen validation ECE threshold. Selecting the most accurate of these on the validation set is a possible use; existing comparisons in Appendix D select by ECE or accuracy separately.

6 CONCLUSION

Standard SFT can improve calibration early and worsen it later when a model is too confident before training. Updates from currently misanswered training questions improve calibration but lose influence as these questions become rarer and their mean gradients shrink relative to initialization. Updates from correctly answered questions continue to raise confidence, while logit scale also affects the outcome. SGCR replays currently misanswered questions and can lower calibration error and extend the calibration window. These results support tracking calibration throughout training alongside task performance.

AI USE STATEMENT

We used OpenAI Codex to assist with polishing and translating text, drafting sections of the manuscript, and retrieving and organizing related literature. AI assistance also supported research ideation and execution, including refining analytical arguments, developing data-analysis and plotting code, and formulating and checking mathematical claims and proofs. We did not use generative AI to create new synthetic training or evaluation datasets for this study. Reported numerical results are derived from recorded experiment artifacts. References and numerical summaries were checked against primary sources and saved experiment records, respectively. The authors take responsibility for the final text, code, analyses, mathematical claims, and results, including all AI-assisted content.

ETHICS STATEMENT

This study uses existing language models, public benchmarks, and existing instruction datasets. MedMCQA is a multiple-choice benchmark; these experiments do not establish clinical reliability. The study includes no new human-participant data collection.

REPRODUCIBILITY STATEMENT

The appendix provides proofs, training configurations, evaluation protocols, and analysis details. The accompanying anonymized code archive contains training and analysis scripts, saved metric summaries, and provenance manifests. Its reproduction guide includes commands for recomputing time-averaged ECE and calibration-window lengths from the included trajectories. Appendix F explains the NLL computation conventions.

REFERENCES

Yu Bai, Song Mei, Huan Wang, and Caiming Xiong. Don't just blame over-parametrization for over-confidence: Theoretical analysis of calibration in binary classification. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/bai21c.html.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of long-form generations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 2732–2778, 2024. URL https://proceedings.mlr.press/v235/band24a.html.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 295–302, 2020. URL https://aclanthology.org/2020.emnlp-main.21/.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784, 2024. doi: 10.18653/v1/2024.emnlp-main.444. URL https://aclanthology.org/2024.emnlp-main.444/.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Guande He, Jianfei Chen, and Jun Zhu. Preserving pre-trained features helps calibrate fine-tuned language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2305.19249.

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

Jerry Huang, Peng Lu, and Qiuhao Zeng. Calibrated language models and how to find them with label smoothing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 25593–25612, 2025. URL https://proceedings.mlr.press/v267/huang25w.html.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. OpenAssistant conversations - democratizing large language model alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/949f0f8f32267d297c2d4e3ee10a2e7e-Abstract-Datasets_and_Benchmarks.html.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/f8c0c968632845cd133308b1a494967f-Abstract.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Beier Luo, Shuoyuan Wang, Sharon Li, and Hongxin Wei. Your pre-trained LLM is secretly an unsupervised confidence calibrator. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ef6b667990416111e3ebbeeba4a8f283-Abstract-Conference.html.

Meta AI. Llama 3.2 model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/aeb7b30ef1d024a76f21a1d40e30c302-Abstract.html.

Preetum Nakkiran, Arwen Bradley, Adam Golinski, Eugene Ndiaye, Michael Kirchhof, and Sinead Williamson. Trained on tokens, calibrated on concepts: The emergence of semantic calibration in LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/39b228b1a83232b14a1c81b0ed5a7a05-Abstract-Conference.html.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://cdn.openai.com/papers/gpt-4.pdf.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2203.02155.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pp. 248–260, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2407.10490.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05952.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model, 2023. URL https://crfm.stanford.edu/2023/03/13/alpaca.html.

Junjiao Tian, Dylan Yung, Yen-Chang Hsu, and Zsolt Kira. A geometric perspective towards neural calibration via sensitivity decomposition. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/dda99de58ff020cfb57fec1404c97003-Abstract.html.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1812.05159.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract.html.

Ziming Wang, Zeyu Shi, Haoyi Zhou, Shiqi Gao, Qingyun Sun, and Jianxin Li. Towards objective fine-tuning: How LLMs' prior knowledge causes potential poor calibration? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14830–14853, 2025. doi: 10.18653/v1/2025.acl-long.722. URL https://aclanthology.org/2025.acl-long.722/.

Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 23631–23644, 2022a. URL https://proceedings.mlr.press/v162/wei22d.html.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022b. URL https://arxiv.org/abs/2109.01652.

Jiancong Xiao, Bojian Hou, Zhanliang Wang, Ruochen Jin, Qi Long, Weijie J. Su, and Li Shen. Restoring calibration for aligned large language models: A calibration-aware fine-tuning approach. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 68364–68390, 2025. URL https://proceedings.mlr.press/v267/xiao25b.html.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

A Extended Related Work

Calibration objectives and logit geometry. Training losses, feature geometry, and calibration are closely connected (Guo et al., 2017; Mukhoti et al., 2020; Tian et al., 2021). Logit normalization controls confidence by constraining logit magnitude (Wei et al., 2022a); analyses of binary classification derive calibration behavior under explicit data and model assumptions (Bai et al., 2021). Our identities measure changes in empirical calibration between LLM checkpoints, and our bounds describe individual updates. Section 4.3 connects scale to the main mechanism: updates from questions the model gets wrong help calibration, but their influence weakens as training proceeds.

Calibration of fine-tuned language models. Language-model calibration depends on the task and on how confidence is measured (Desai & Durrett, 2020; Jiang et al., 2021; Kadavath et al., 2022). Feature preservation, label smoothing, and calibration-aware SFT offer ways to improve it (He et al., 2023; Huang et al., 2025; Xiao et al., 2025). Calibration constraints can also depend on how examples relate to the model's existing knowledge (Wang et al., 2025). Disagreement with a pretrained model can inform post-hoc calibration (Luo et al., 2025). We study standard NLL training and test its calibration effects by rescaling logits and changing the fraction of incorrectly answered examples in a batch. SGCR keeps NLL and repeatedly samples examples the model currently answers incorrectly.

Learning dynamics and example selection. Analyses of LLM fine-tuning relate learning to the model's knowledge and to changes in its output distribution (Gekhman et al., 2024; Ren & Sutherland, 2025). Studies of example forgetting track when predictions change from correct to incorrect and use these histories to guide data selection (Toneva et al., 2019). Prioritized experience replay gives informative transitions more updates in reinforcement learning (Schaul et al., 2016). SGCR samples questions the model currently answers incorrectly to retain training updates that can improve calibration. The interventions in Section 5.1 test one update, and the windows in Section 5.3 measure how long repeated sampling keeps ECE below a threshold. Screening cost and the scope of the replay comparison are given in Appendix D.

How confidence is measured. Linguistic calibration (Band et al., 2024) and semantic calibration theory (Nakkiran et al., 2026) address confidence in richer forms of generated text. Our answer-token results do not establish that generated explanations are calibrated. Appendices M and O retain the full-response SFT experiments and earlier controls, including accuracy losses and negative results. The calibration window measures how long observed ECE meets a chosen threshold under fixed bins. It does not certify population calibration or calibration at unobserved training steps.

B Proofs and Conditions for One-Step Predictions

This section derives the results used in the main text. The first-order expansion of held-out logit changes follows existing learning-dynamics analysis (Ren & Sutherland, 2025). The increase in confidence with logit norm is also established (Wei et al., 2022a). We use these results to explain what the experiments measure. The ECE decomposition is an exact identity; the update bounds apply to one step under the stated conditions. They do not specify the full training trajectory of an LLM optimized with AdamW.

B.1 Training and evaluation probabilities

Training logits $\ell_i \in \mathbb{R}^V$ include every vocabulary token. Evaluation logits $z_x = HS_x\ell_x \in \mathbb{R}^K$ are centered over the legal answer candidates. H and S_x are fixed for each prompt. The Jacobians $J_i \in \mathbb{R}^{V\times d}$ and $J_x^c \in \mathbb{R}^{K\times d}$ are evaluated at the same parameter values. Vector norms are Euclidean unless otherwise specified; matrix norms in Lipschitz assumptions are induced operator norms. The derivative calculations assume real arithmetic and differentiable logits. The BF16 experiments test how well the approximations describe actual updates. An expectation over a training group denotes its arithmetic mean, or its fixed sampling-distribution mean when explicitly specified.

The differentiable objective $\mathcal{R}$ is a function of the held-out candidate logits with evaluation labels fixed. Its output gradient $\beta_x = \partial\mathcal{R}/\partial z_x$ already contains any $1/n$ normalization. Therefore the chain

rule gives

$$h = \nabla_\theta \mathcal{R} = \sum_x (J_x^c)^\mathsf{T} \beta_x. \tag{12}$$

When differentiating the smooth calibration objective, we hold the predicted answers and their correctness indicators fixed at the checkpoint. This makes the chosen objective differentiable; it does not make every implementation of hard-binned ECE differentiable.

B.2 How logit scale changes the NLL gradient

Lemma 1 (Residual magnitude under scaling). *For $p = \mathrm{softmax}(\ell/\tau)$ and one-hot target e_y,*

$$\nabla_\ell[\tau \,\mathrm{CE}(\ell/\tau, y)] = p - e_y, \qquad \|p - e_y\|_1 = 2(1 - p_y). \tag{13}$$

More specifically, let $p(a) = \mathrm{softmax}(av)$ for $a > 0$. If y is the unique maximum of v with margin $m = \min_{j \neq y}(v_y - v_j) > 0$, then

$$1 - p_y(a) \leq (V - 1)e^{-am}. \tag{14}$$

If some j instead satisfies $v_j - v_y \geq \delta > 0$, then $p_y(a) \leq e^{-a\delta}$.

Proof. The chain rule cancels τ with $1/\tau$. Since $0 \leq p_y \leq 1$ and $\sum_{j \neq y} p_j = 1 - p_y$, the absolute residual entries sum to $2(1 - p_y)$. In the first margin case,

$$1 - p_y = \frac{\sum_{j \neq y} e^{-a(v_y - v_j)}}{1 + \sum_{j \neq y} e^{-a(v_y - v_j)}} \leq \sum_{j \neq y} e^{-am} = (V - 1)e^{-am}.$$

In the second case, the softmax denominator contains e^{av_j}, so $p_y \leq e^{a(v_y - v_j)} \leq e^{-a\delta}$. □

Increasing scale can make the residual small for a target that already has high probability, while leaving a large residual when the model predicts an incorrect token. The lemma requires a margin over the full vocabulary. An incorrect candidate prediction guarantees that a non-target vocabulary logit exceeds the target logit. A correct candidate prediction does not guarantee that the target is the vocabulary-wide maximum. Appendix H therefore measures each group's share of residual norms using the actual full-vocabulary probabilities and sums of ℓ_2 norms. It does not substitute candidate confidence into the lemma. The output residual norm also does not determine the parameter-gradient norm $\|J_i^\mathsf{T} \epsilon_i\|$ or its alignment with the calibration gradient.

B.3 Proof of Proposition 1

Proof. Set $d = b + Pg$ and $\theta^+ = \theta - \eta d$. The fundamental theorem of calculus gives

$$z_x(\theta^+) - z_x(\theta) = -\eta J_x^c(\theta) d - \eta \int_0^1 \left[J_x^c(\theta - v\eta d) - J_x^c(\theta)\right] d \, dv.$$

The second term has norm at most $\eta \int_0^1 L_x v\eta \|d\|^2 \, dv = L_x \eta^2 \|d\|^2/2$. Substituting $g = \sum_i w_i J_i^\mathsf{T} \epsilon_i$ into the first term yields Equation 4. Dimensions are $K_P(x, i) \in \mathbb{R}^{K \times V}$ and $K_P(x, i)\epsilon_i \in \mathbb{R}^K$. □

If the optimizer map is nonlinear, the valid expansion is $\Delta z_x = -\eta J_x^c d_\omega(g) + e_x$, with the same form of remainder. In general, a fixed matrix P cannot express this update as a sum over examples. For an affine map, substituting Equation 12 gives

$$A_G = h^\mathsf{T} P g_G = \sum_x \beta_x^\mathsf{T} \, \mathbb{E}_{i \in G}[K_P(x, i)\epsilon_i], \qquad \beta_x = \nabla_{z_x} \mathcal{R}. \tag{15}$$

Here $A_G = h^\mathsf{T} P g_G$ and $\beta_x = \nabla_{z_x} \mathcal{R}$. The measured gradient projection summarizes the effect on held-out calibration with one scalar; it does not recover the full matrix that maps training residuals to held-out logit changes.

B.4 How norm and direction change confidence

For $r > 0$ and a unique winning candidate k, the softmax Jacobian gives

$$\nabla_z c = c(e_k - q), \qquad \dot{r} = u^\mathsf{T}\dot{z}, \qquad \dot{u} = \frac{(I - uu^\mathsf{T})\dot{z}}{r}.$$

Substituting $\dot{z} = u\dot{r} + r\dot{u}$ gives

$$\dot{c} = c\underbrace{(u_k - q^\mathsf{T}u)}_{>0}\dot{r} + cr(e_k - q)^\mathsf{T}\dot{u}. \tag{16}$$

Its radial coefficient is

$$u_k - q^\mathsf{T}u = \sum_{j \neq k} q_j(u_k - u_j) > 0. \tag{17}$$

Every q_j is positive at finite logits and the winner is unique. Consequently, at fixed direction, confidence strictly increases with radius, while the predicted answer is unchanged. Along an arbitrary learning trajectory the directional term has no fixed sign. A growing *mean* norm alone also need not imply growing mean confidence, since radial coefficients and individual changes differ across examples.

Combining this derivative with Proposition 1 gives the first-order change in logit norm:

$$\Delta r_x = -\eta u_x^\mathsf{T} J_x^c b - \eta \sum_i w_i u_x^\mathsf{T} K_P(x, i)\epsilon_i + O(\eta^2), \tag{18}$$

away from $r_x = 0$, with bounded derivatives and a bounded update direction. Whether the norm decreases depends on how the training residuals affect held-out logits and how those changes project onto u_x. Residual size alone does not prove that the norm decreases early or increases later. We measure these changes in the saved trajectories and test the effect of scale through interventions.

B.5 Proof of Proposition 2

Proof. The two symmetric terms telescope pointwise:

$$A_r + A_u = \tfrac{1}{2}[(F_{ts} - F_{ss}) + (F_{tt} - F_{st}) + (F_{st} - F_{ss}) + (F_{tt} - F_{ts})] = F_{tt} - F_{ss} = c_t - c_s.$$

For a single checkpoint, $|d_j| = d_j + 2(-d_j)_+$ and the bins partition the evaluation set. Hence

$$\mathrm{ECE} = \sum_j |d_j| = \sum_j d_j + 2\sum_j (-d_j)_+ = \bar{c} - \bar{a} + 2U.$$

Taking the difference between checkpoints and substituting the first identity proves Equation 8. Each endpoint can have different winning candidates and bin assignments; only the evaluation examples and the definition of ECE are shared. □

The same argument gives $\mathrm{ECE} \geq |\bar{c} - \bar{a}|$. Equality with the *signed* gap requires $U = 0$, meaning no negatively signed nonempty bin. Small signed gap alone is insufficient evidence of calibration, because positive and negative bin gaps can cancel. For example, two equally weighted bins with within-bin confidence–accuracy gaps $+0.2$ and -0.2 have zero global gap and ECE 0.2.

The decomposition averages the two orders of changing norm and direction. It treats the two factors symmetrically, assigns zero to a factor that makes no difference in either order, and sums to the observed confidence change. These properties do not establish a unique causal interpretation. The intermediate logits $r_s u_t$ and $r_t u_s$ need not be produced by the training algorithm. At a zero-norm endpoint, any centered unit direction can be used to form these combinations. Their contributions still sum to the confidence change, but the individual terms depend on the chosen direction. Every evaluated logit vector has nonzero norm, so this ambiguity does not affect the reported results.

B.6 When hard-binned ECE can be differentiated

Lemma 2 (ECE within a fixed prediction and bin region)**.** *Suppose all evaluation winners are unique, all confidences avoid bin boundaries, and each nonempty bin has a nonzero signed gap at θ. With continuous differentiable logits, these assignments and signs are fixed in a sufficiently small neighborhood. In that neighborhood,*

$$\nabla_{z_x} \mathrm{ECE} = \frac{\mathrm{sign}(d_{j(x)})}{n}\, c_x(e_{k_x} - q_x). \tag{19}$$

If every bin is overconfident, this is the gradient of mean confidence.

Proof. Strict inequalities at finitely many evaluation examples persist in a sufficiently small neighborhood. Correctness indicators are then constant and each absolute bin gap can be differentiated with its fixed sign. Differentiating d_j and applying the softmax derivative gives Equation 19. □

The lemma describes a neighborhood in which predictions and bin assignments remain fixed. A finite LLM update may leave that neighborhood. The exact identity in Proposition 2 still holds when assignments change, but a smoothness bound for hard ECE requires the entire update to stay within a suitable smooth region. Our gradient experiments use the smooth objective defined in Appendix J; hard ECE is measured after the update.

B.7 Proof of Proposition 3

Proof. Let $d = d_\omega(g_\alpha)$. An integral expansion of $\mathcal{R}$ gives

$$\mathcal{R}(\theta - \eta d) - \mathcal{R}(\theta) = -\eta h^\mathsf{T} d - \eta \int_0^1 [\nabla \mathcal{R}(\theta - v\eta d) - h]^\mathsf{T} d\, dv.$$

Lipschitz continuity of the gradient bounds the absolute integral term by $L\eta^2\|d\|^2/2$. The stated strict inequality makes the sum negative. If $d = b + Pg_\alpha$, linearity yields

$$h^\mathsf{T} d = M + \alpha A_W + (1-\alpha) A_C.$$

Its negative, multiplied by η, has slope $-\eta(A_W - A_C)$ and the stated root. If the root lies outside $[0, 1]$, the first-order term does not change sign within the interval. If $A_W = A_C$, changing the Wrong fraction has no first-order effect in this affine model. □

The threshold is where the first-order term equals zero; the actual finite update need not have zero effect there. For the affine case with $D = A_W - A_C > 0$, define $B_\alpha = L\eta\|d_\alpha\|^2/2$. Improvement is guaranteed if $M + A_C + \alpha D > B_\alpha$, and worsening is guaranteed if $M + A_C + \alpha D < -B_\alpha$. The bound does not determine the sign between these margins. If the finite update has a zero and $\|d_\alpha\| \le D_0$ over the interval, that zero is at most $L\eta D_0^2/(2D)$ from α_*. This requires the same smooth objective, a fixed affine map, and known constants. It does not provide an error bar for the empirical AdamW threshold plot.

B.8 How the optimizer affects the result

Where the optimizer map is differentiable, set $v = g_W - g_C$ and $\theta_\alpha^+ = \theta - \eta d_\omega(g_\alpha)$. The exact derivative of the post-update objective is

$$\frac{d}{d\alpha}\mathcal{R}(\theta_\alpha^+) = -\eta \nabla \mathcal{R}(\theta_\alpha^+)^\mathsf{T} D d_\omega(g_\alpha) v. \tag{20}$$

Replacing the gradient at θ_α^+ with h incurs absolute error at most $L\eta^2\|d_\omega(g_\alpha)\|\|Dd_\omega(g_\alpha)v\|$. To determine whether increasing the Wrong fraction consistently improves calibration, we therefore need the derivative of the optimizer map as well as the calibration gradient. Active clipping adds another derivative and a boundary that requires a one-sided calculation.

A larger aligned gradient can produce less improvement under Adam. Consider one coordinate, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and previous moments $m \simeq 0.1$, $v \simeq 0.01$. Such moments follow 1,000 previous gradients all equal to 0.1 from zero initialization. With current gradient g and negligible ε, the bias-corrected Adam direction is, to the displayed precision,

$$f(g) = \frac{0.09 + 0.1g}{\sqrt{0.0095 + 0.05g^2}}.$$

It satisfies $f(0.3) = 1.0142$ and $f(0.4) = 0.9827$. Both gradients are positive and below a clipping norm of one, yet the larger gradient produces the smaller descent step. For a binary evaluation logit $\theta > 0$ with the wrong winning answer, empirical ECE is $\sigma(\theta)$. Starting at $\theta = 1$ with learning rate 0.01, the larger gradient therefore gives a smaller ECE reduction, although its unpreconditioned calibration alignment is larger. Decoupled weight decay at the same θ adds a common direction and does not remove this ordering.

More generally, apart from positive bias-correction factors, one coordinate has the form $f(g) = (M + Cg)/(\sqrt{V + Dg^2} + \varepsilon)$ for $V > 0$. Direct differentiation yields

$$f'(g) = \frac{CV + C\varepsilon\sqrt{V + Dg^2} - DMg}{\sqrt{V + Dg^2}(\sqrt{V + Dg^2} + \varepsilon)^2},$$

which can be negative. Thus, $h^\mathsf{T}(g_W - g_C) > 0$ alone does not guarantee that increasing the Wrong fraction improves a native Adam update. This example establishes that limitation; it is not fitted to the LLM optimizer and concerns the threshold comparisons in Figure 17e.

Sampled batches and an entire training path. The controlled gradient mixture in Proposition 3 is deterministic. For a sampled batch B with $\mathbb{E}g_B = g_\alpha$, a nonlinear update generally satisfies $\mathbb{E}[d_\omega(g_B)] \neq d_\omega(g_\alpha)$. The batch-averaged leading effect is $-\eta h^\mathsf{T}\mathbb{E}[d_\omega(g_B)]$, with an expected remainder bounded by $L\eta^2\mathbb{E}\|d_\omega(g_B)\|^2/2$ when the same smoothness bound holds. Sampling without replacement changes its variance, not this distinction.

During training, g_W, g_C, π_W, h, J, and the optimizer moments all change. Even an exact derivative at one checkpoint cannot identify a future minimum without assumptions about these later changes. The experiment that predicts the minimum from the first zero crossing does not impose or validate such assumptions. Repeated crossings and the observed timing errors show that this test alone cannot determine when to stop training.

Independent numerical checks. The archive includes finite-difference checks of the confidence derivative with respect to norm, the $O(\eta^2)$ remainder for held-out logit changes on a nonlinear toy model, the one-step bound, and the Adam counterexample with exact finite-history moments and bias corrections. It also verifies both decomposition identities at every saved evaluation checkpoint. These numerical checks test the algebra and implementation; the proofs are given above.

C Additional Measurements of Logit Norm and Direction

Section 4.3 summarizes the norm–direction decomposition. Figure 7 retains the full saved-logit analysis. The mean norm falls from 21.91 before training to 3.16 at step 39 and later reaches 6.03. In the fixed initially correct group, the gold logit rises from 2.79 at step 39 to 5.02 at the endpoint. In the fixed initially wrong group, the original wrong option remains above the gold answer: their mean logits rise from 0.99 and 0.38 at step 390 to 1.92 and 0.56 at step 1,170. Later training strengthens originally favored wrong answers as well as correct ones.

The analysis uses centered candidate logits for the same question IDs at every checkpoint. For each seed, the gold margin is $z_y - \max_{j \neq y} z_j$. Figure 8 shows its cumulative distribution within fixed initial correctness groups. A negative margin means the correct answer remains below a distractor. Keeping the population fixed prevents changes in group membership from being mistaken for within-question learning.

From initialization to step 39, mean confidence decreases by 23.26 percentage points. The symmetric decomposition assigns -20.83 points to norm and -2.43 to direction; accuracy rises by 1.11 points.

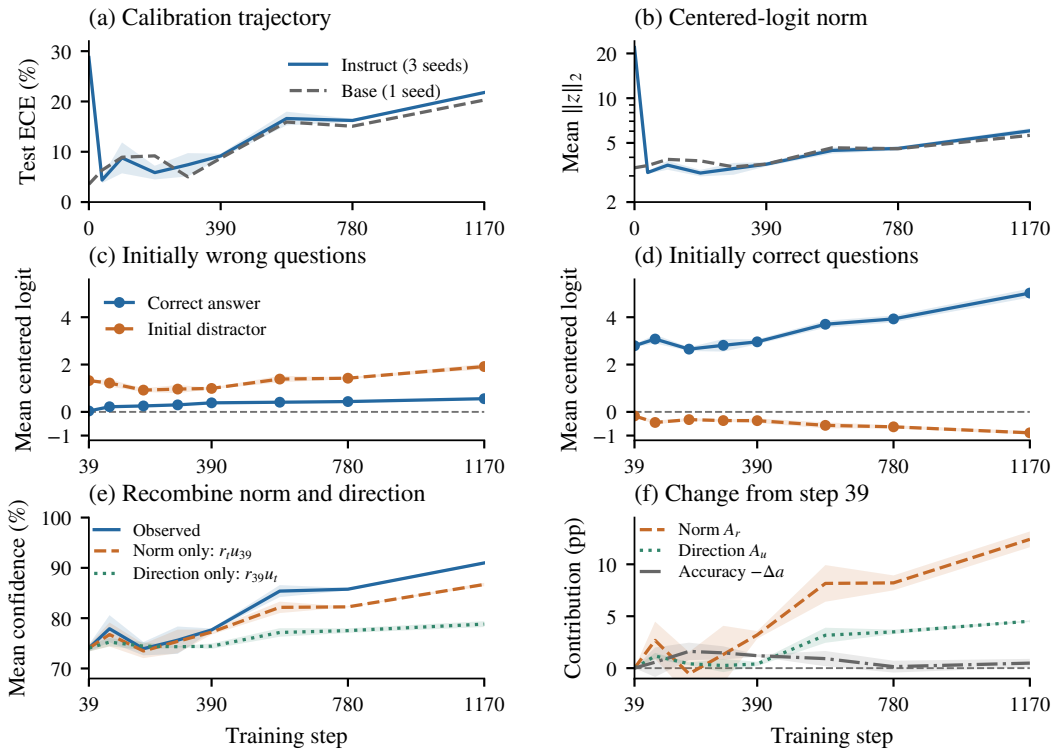


Figure 7: **Increasing logit norm accounts for most of the late confidence increase.** (a,b) Test ECE-10 and mean centered-logit norm. (c,d) Gold and strongest initial distractor logits for fixed initially wrong/correct question sets. (e) Observed confidence and predictions computed with direction or norm held at step 39. (f) Norm, direction, and accuracy terms of Equation 8, relative to step 39. The remaining $2\Delta U$ term is below 0.04 percentage points in magnitude. Instruct curves show three-seed mean $\pm$ SD; Base uses one seed. Panels (c–f) begin after the initial decrease in norm; distributions and the initial-stage decomposition appear in Appendix C.

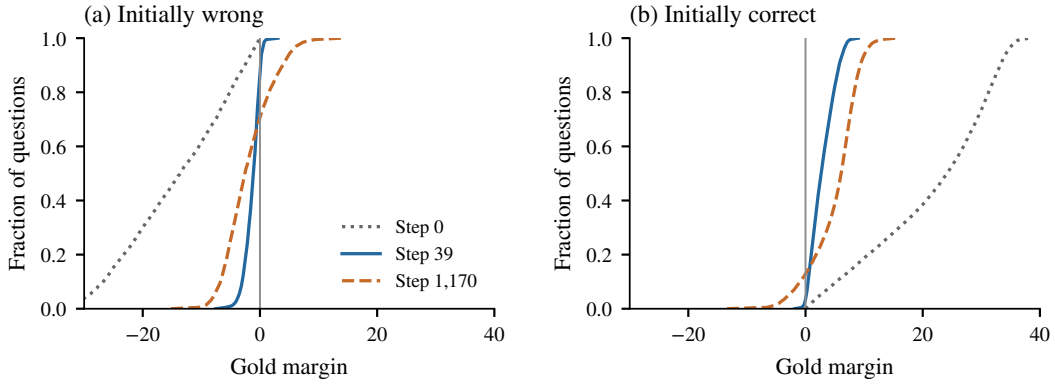


Figure 8: Gold-margin distributions for fixed initially wrong/correct MMLU test questions, Qwen Instruct seed 1. Curves use 201 empirical quantiles from each full group; the horizontal display range is restricted to $[-30, 40]$. The vertical line marks the decision boundary. The strongest distractor in this margin is recomputed at each state, whereas Figure 7c,d holds the initial distractor fixed.

From step 39 to 1,170, the corresponding terms are $+12.40$, $+4.52$, and -0.48 points of accuracy change. Averaging the two orders accounts for interactions between norm and direction. Simply subtracting the curves obtained by holding one factor fixed does not give additive effects. Across all

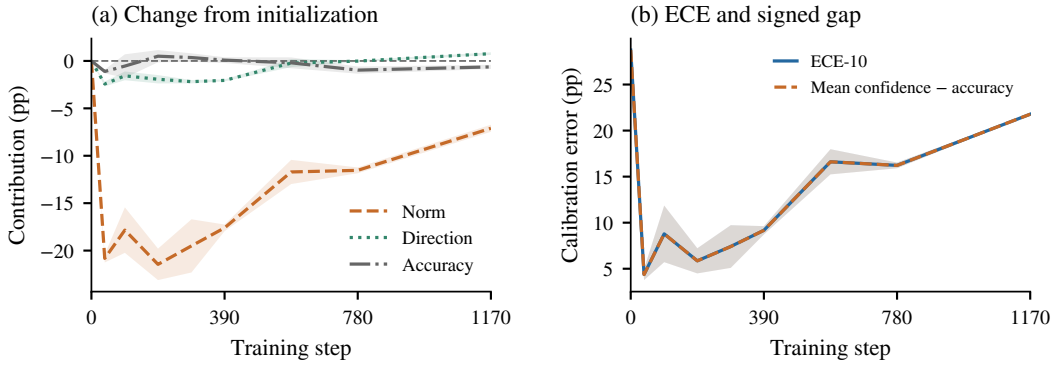


Figure 9: **The exact decomposition also explains the early improvement.** (a) Mean norm, direction, and negative accuracy-change terms, relative to initialization. (b) ECE-10 and the mean confidence–accuracy gap nearly coincide because underconfidence is small. Lines and bands show three-seed mean $\pm$ SD. The terms are computed by combining saved logits, rather than by intervening on training.

27 Instruct checkpoints, $2U \leq 0.0335$ percentage points. The discrepancy in the exact identity is below 10^{-12} in probability units. No additional model training or inference is involved.

D Calibration Windows and State-Gated Corrective Replay

D.1 Compared runs and evaluation steps

The SGCR comparison is an independent Qwen3-4B Instruct–MMLU training pair with seed 1. It is separate from the three-seed trajectories in Figures 1 and 2; the Vanilla runs and their endpoints are not interchangeable. Both methods use the same 99,842 training examples, 1,531 validation questions, and 14,042 test questions. Answer-token full-vocabulary NLL, global batch size 256, and the planned 1,170-update schedule are matched. Within the common horizon, the recorded learning rates agree at every evaluation step.

The preserved snapshot is from September 10, 2026, 04:31 UTC. Vanilla has completed all 1,170 updates and has 37 completed evaluations. SGCR has completed 449 updates, with 14 completed evaluations through step 429. The paired analysis uses exactly the common observed steps

$$0,\ 39,\ 78,\ 98,\ 117,\ 156,\ 195,\ 234,\ 273,\ 293,\ 312,\ 351,\ 390,\ 429.$$

No later SGCR evaluation is imputed. The source report verifies the saved prediction hashes, fixed evaluation denominators, and exact alignment of metrics to checkpoints. We reproduce its table field by field before computing windows. The longer Vanilla trajectory remains available in the accompanying data but is not used to give Vanilla a different observation horizon in the window comparison.

D.2 Which examples are replayed and how much training they receive

Each optimizer update contains 256 unique training examples. At least 230 are ordinary uniform samples; at most 26 positions are reserved for corrective replay, a maximum replay fraction of $26/256 = 10.15625\%$. A full-training-set pass every 0.25 epoch refreshes a pool of Wrong candidates. Before an update, replay candidates are checked with the current model in the actual training context. Candidates that have become Correct are replaced from a finite randomized candidate queue. Accepted replay samples are verified as Wrong in the gradient forward pass. If eligible unique candidates are insufficient, ordinary uniform samples fill the remaining positions.

A rejected candidate can be replayed later if the model answers it incorrectly again. Examples outside the current pool can enter when the pool is refreshed. Correctness uses the same answer-option argmax as the batch audit. Uniformly sampled examples can also be Wrong, so the replay quota differs from the total Wrong fraction in the final batch. Replay changes neither NLL, gradient

averaging, nor the native AdamW update. Screening requires additional forward passes; matching optimizer steps therefore does not match wall time or total compute.

The snapshot records 11,393 realized replay positions across 449 updates, no accepted replay sample that was Correct at its actual training forward, and 15 updates with uniform fallback. These counts audit adherence to the sampling rule.

D.3 Measuring window length at different thresholds

Equation 11 uses consecutive *observed* evaluations. A window ends at its last qualifying checkpoint, rather than at an interpolated threshold crossing. Its duration is measured in optimizer steps, not in the number of saved checkpoints. An isolated qualifying checkpoint has zero observed span. When multiple windows share the maximum span, the earliest is displayed; all qualifying intervals are retained in the released table. Windows that reach the final observation are marked as right-censored. These conventions describe the evaluation grid and do not certify calibration at unobserved intermediate updates.

The 5% threshold is an illustrative retrospective choice, not a preregistered operating requirement. Table 3 gives several thresholds, and Figure 6b shows the exact step function in threshold: window length changes only when the threshold crosses an observed ECE value. An independent enumeration of all contiguous checkpoint subsequences verifies every reported length. No curve smoothing merges separated windows.

At 5%, SGCR has an early window at steps 39–78 and a longer window at 117–293. Its step-98 ECE of 5.60% separates these intervals. Vanilla's longest window is 78–98. The first subsequent above-threshold evaluations occur at step 312 for SGCR and step 117 for Vanilla; the reported lengths use the last qualifying evaluations, 293 and 98. The validation split yields the same longest-window endpoints. Validation and test agreement is measured on separate question sets, but both come from the same pair of training runs.

Table 3: Longest observed calibration windows within the shared steps 0–429. Endpoints are qualifying saved checkpoints; lengths are optimizer-step spans. Validation and test are measured on different questions from the same training pair. These thresholds are retrospective sensitivity choices.

Split	ε (%)	Vanilla interval	$L_\varepsilon \uparrow$	SGCR interval	$L_\varepsilon \uparrow$
Test	4	78–98	20	117–293	176
Test	5	78–98	20	117–293	176
Test	6	39–98	59	39–293	254
Test	7.5	39–117	78	39–293	254
Test	10	39–117	78	39–390	351
Validation	4	78–98	20	117–293	176
Validation	5	78–98	20	117–293	176
Validation	6	78–117	39	39–293	254
Validation	7.5	39–117	78	39–293	254
Validation	10	39–117	78	39–429†	390

† The window reaches the observation horizon; its later endpoint is unobserved.

D.4 Accuracy, Brier score, NLL, and Wrong examples

Figure 10 reports calibration, accuracy, proper scores, and supervision composition over the complete common observation interval. The mean test ECE obtained by trapezoidal integration over steps 0–429 is 9.68% for Vanilla and 5.43% for SGCR. This descriptive area linearly connects saved metric values; the window calculation itself does not interpolate. Within the respective longest 5% windows, observed test accuracy ranges from 70.06–70.33% for Vanilla and 67.80–69.23% for SGCR. Longer calibration windows therefore do not imply equal task performance.

Table 4 compares validation-selected checkpoints with the common endpoint. Vanilla's lowest-validation-ECE checkpoint is step 78, with test accuracy 70.33% and ECE 3.35%. SGCR's corresponding checkpoint is step 156, with accuracy 68.50% and ECE 1.48%. At step 429, SGCR has

ECE 10.72% versus 11.36%, but accuracy 68.04% versus 68.60%, Brier 0.4508 versus 0.4407, and candidate NLL 0.9934 versus 0.8821. The early Vanilla checkpoint retains higher accuracy than the SGCR checkpoint selected by lowest validation ECE.

Table 4: **Checkpoint choices trade off calibration and task performance.** Qwen–MMLU test outcomes at validation-selected checkpoints and the common endpoint. Selection uses the available shared horizon, with ties resolved by earliest step. The two selection criteria are applied separately and retrospectively. Brier and candidate NLL retain the source evaluation's conventions.

Method	Checkpoint criterion	Step	Acc. (%) ↑	ECE-10 (%) ↓	Brier ↓	NLL ↓
Vanilla	Lowest val. ECE	78	70.33	3.35	0.3951	0.7529
Vanilla	Highest val. acc.	98	70.06	3.82	0.3975	0.7576
Vanilla	Common endpoint	429	68.60	11.36	0.4407	0.8821
SGCR	Lowest val. ECE	156	68.50	1.48	0.4144	0.7866
SGCR	Highest val. acc.	78	69.90	2.74	0.3958	0.7518
SGCR	Common endpoint	429	68.04	10.72	0.4508	0.9934

At step 429, the Wrong fraction over the full training set is 1.39% for SGCR and 5.15% for Vanilla. Their next actual training batches contain 12.50% and 7.42% Wrong examples, respectively. The full-set fractions describe predictions at checkpoint θ_{429}; the batch fractions are measured during the gradient forward pass for update 430. Replay thus increases how often current errors enter an update, even as they become less common in the training set. These counts do not measure the group-gradient norm or alignment in Equation 5. Those quantities and the changing optimizer state can still reduce the calibration benefit.

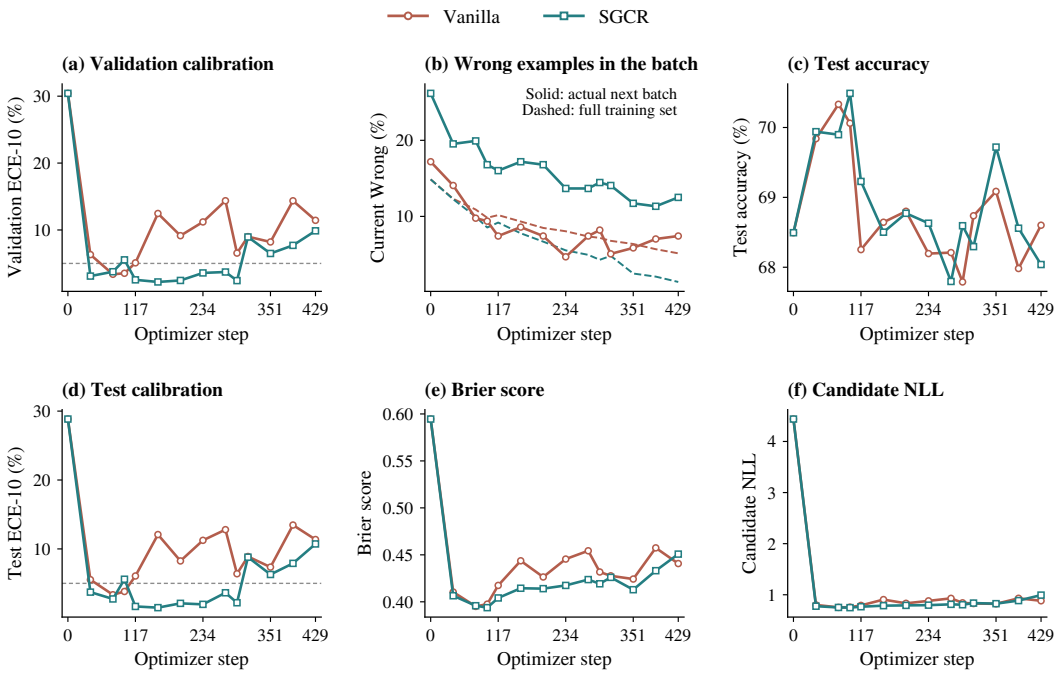


Figure 10: **Paired trajectories behind the calibration-window comparison.** One seed, common observations through step 429. (a,d) Validation and test ECE-10; the dashed horizontal line is 5%. (b) Solid curves measure current Wrong exposure in the next actual gradient batch; dashed curves measure the full-training-set Wrong fraction at the checkpoint. (c,e,f) Test accuracy, Brier, and candidate NLL. Lines connect observed evaluations; no seed uncertainty is inferred from checkpoint variation.

The observed window extension is measured retrospectively from this single pair. Complete paired runs, independent seeds, and a replay control with the same quota but no correctness recheck would

assess consistency and the effect of the recheck. Section 5.3 reports the measured task tradeoffs and distinguishes the existing checkpoint comparisons from the proposed threshold-based use.

D.5 Calibration windows across models and tasks

Three additional paired comparisons extend the calibration-window analysis to Llama-3.2-3B on MMLU and to Qwen3-4B and Llama-3.2-3B on MedMCQA. SGCR has longer observed windows on Llama–MMLU and Qwen–MedMCQA at the illustrative 5% threshold. Llama–MedMCQA shows longer windows at broader thresholds, while retaining only an isolated qualifying checkpoint at 5%.

Each pair uses its own matched Vanilla baseline, seed 1, and the same 14 observation steps through update 429. Both methods use single-letter A/B/C/D responses with an empty answer prefix, rather than the JSON-answer interface of the original Qwen–MMLU pair above. Two B200 GPUs train with microbatch size four and 32 accumulation steps per GPU (global batch size 256); a third GPU evaluates saved checkpoints. The original learning-rate schedules retain horizons of 1,170 updates for MMLU and 2,094 for MedMCQA; 429 updates correspond to 1.10 and 0.615 epochs, respectively. This common observation horizon was chosen after earlier exploratory results, rather than by an online stopping rule.

MMLU uses 99,842 training examples, 1,531 validation questions, and 14,042 separate labeled test questions. MedMCQA uses 178,726 training examples and its 4,183-question official validation split. No independent labeled MedMCQA test results are available in this comparison. All window calculations use the same ECE-10 definition and consecutive-observation convention as Equation 11.

Time-averaged calibration error. We summarize ECE over updates 0–429 by integrating the observed trajectory with the trapezoidal rule and dividing by the 429-update horizon. The average weights each interval by its duration rather than weighting checkpoints equally. Llama–MMLU test ECE decreases from 13.55% for Vanilla to 7.10% for SGCR, a 47.6% relative reduction. Official-validation ECE decreases from 7.36% to 4.09% for Qwen–MedMCQA and from 14.49% to 9.27% for Llama–MedMCQA, relative reductions of 44.5% and 36.0%, respectively. These are summaries of matched single-seed trajectories, not averages across seeds or tasks.

The same ECE thresholds for each comparison. Table 5 reports the same threshold grid for every pair, and Figure 11 shows the exact threshold dependence. At 5%, Llama–MMLU has an SGCR test window at steps 195–273 (78 steps), while Vanilla has no qualifying checkpoint. The corresponding SGCR validation window is 195–234 (39 steps). On Qwen–MedMCQA, the longest window increases from steps 39–156 for Vanilla to 39–429 for SGCR, or 117 to 390 steps. SGCR remains below the threshold at the observation horizon, so its later endpoint is unobserved. For these two pairs, SGCR's window is longer throughout thresholds of 4–10% on every available evaluation split.

Table 5: Longest observed window lengths ↑ in optimizer steps for the additional pairs. Every method uses the same observation grid and horizon 429. Thresholds are shared absolute ECE-10 values.

			ε (%)				
Model–task	Split	Method	4	5	6	7.5	10
Llama–MMLU	Test	Vanilla	0	0	0	0	0
		SGCR	39	78	78	98	234
	Validation	Vanilla	0	0	0	0	0
		SGCR	39	39	39	98	156
Qwen–MedMCQA	Validation	Vanilla	78	117	117	117	254
		SGCR	390†	390†	390†	390†	390†
Llama–MedMCQA	Validation	Vanilla	0	0	0	0	0
		SGCR	0†	0†	19	39	117

† The displayed window reaches step 429; its later endpoint is unobserved. Zero denotes no qualifying checkpoint or an isolated point. For Llama–MedMCQA SGCR at 4% and 5%, only step 429 qualifies. MedMCQA has no independent test evaluation here.

Llama–MedMCQA has no positive-span window at 5%: only SGCR's final checkpoint qualifies. Broader thresholds yield SGCR windows of 19 steps at 6% (293–312), 39 steps at 7.5% (195–234), and 117 steps at 10% (195–312). Vanilla has no qualifying checkpoint at any of these thresholds; its minimum validation ECE is 10.52%. SGCR's positive window-length difference persists throughout 6–10%. The shared threshold grid retains the stricter 5% result alongside these broader-threshold gains.

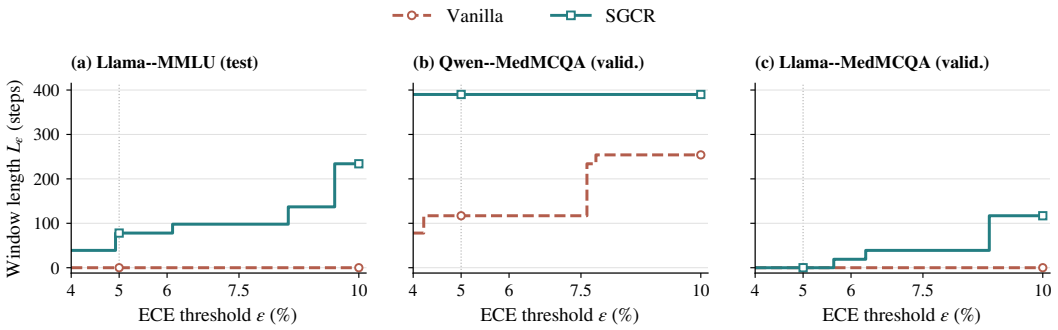


Figure 11: **Window length at different ECE thresholds.** Exact step functions from the common 429-update horizon, with one seed per pair. Markers identify 5% and 10%. MMLU uses separate test questions; MedMCQA panels use official validation. Crossing times are not interpolated, and no uncertainty bands are inferred from these single runs.

Previously recorded relative thresholds. The experiment records also retain a shared threshold for each pair equal to its lowest observed Vanilla validation ECE plus two percentage points. Table 6 applies those values unchanged to both methods. The MMLU test comparison uses the same validation-derived threshold. SGCR has longer validation windows in all three pairs under this convention. These baseline-relative thresholds describe persistence near each baseline's calibration floor; they differ from a common absolute ECE requirement and are computed from the observed Vanilla trajectory.

Table 6: Windows under the previously recorded threshold $\varepsilon_{\text{rel}} = \min_k \text{ECE}_{10}^{\text{Vanilla,valid}}(\theta_{t_k}) + 0.02$. The same full-precision value is used for both methods; displayed percentages are rounded.

			Vanilla		SGCR	
Model–task	Split	ε_{rel} (%)	Interval	Length ↑	Interval	Length ↑
Llama–MMLU	Validation	10.91	39	0	156–390	234
	Test	10.91	39–98	59	156–390	234
Qwen–MedMCQA	Validation	3.19	39–98	59	293–429†	136
Llama–MedMCQA	Validation	12.52	195‡	0	195–429†	234

Lengths are optimizer-step spans. † The later endpoint is unobserved beyond step 429. ‡ Steps 195 and 351 are isolated qualifying points; the earliest zero-span tie is displayed. Thresholds are descriptive, not preregistered operating requirements.

These single-seed comparisons extend the window result across models and tasks, with gains that depend on the threshold. Section 5.3 reports task performance alongside the windows. The accompanying analysis retains every qualifying interval, every tie for the longest window, and each threshold at which the window length changes. Independent seeds and held-out confirmation are needed to assess how consistently the effect occurs.

D.6 Longer Qwen–MedMCQA training

The Qwen3-4B–MedMCQA pair above is continued from each method's step-429 checkpoint to a matched endpoint of 698 updates (one epoch), adding 269 updates per method. The original 2,094-update learning-rate horizon and 209-update warmup remain unchanged, as do seed 1, global batch size 256, answer-token NLL, and the SGCR sampling rule. Seven evaluations at steps 468, 507,

546, 585, 624, 663, and 698 extend the original 14-point grid. A repeated evaluation at the restoration boundary verifies agreement but is not counted as an additional observation. All results use the same 4,183 official validation questions; no independent labeled MedMCQA test set is used.

Window duration and interruption. Table 7 retains the absolute thresholds and the previously recorded relative threshold from the 429-update analysis. At 5%, SGCR's longest window extends from 39–429 (390 steps, right-censored) to 39–468 (429 steps), while Vanilla's remains 39–156 (117 steps). SGCR's next evaluation at step 507 has ECE 8.01%, resolving the formerly unobserved end of this qualifying run. The observed span is 312 steps longer than Vanilla's, or 3.67 times its length. The subsequent qualifying SGCR checkpoints at 546 and 698 are isolated at this threshold and are not joined to the earlier window. The latter isolated point is right-censored with zero observed span. At 10%, SGCR's 39–698 window remains right-censored. None of these spans certifies that unmeasured intermediate updates meet the threshold.

Table 7: **Calibration windows after extending Qwen–MedMCQA to 698 updates.** Longest consecutive observed intervals and optimizer-step spans on official validation, seed 1. A dagger marks a right-censored longest interval. The relative threshold is frozen at the original Vanilla minimum ECE plus two percentage points; it is not recomputed from the continuation.

	Vanilla		SGCR	
ECE threshold	Interval	Span ↑	Interval	Span ↑
4%	39–117	78	39–468	429
5%	39–156	117	39–468	429
6%	39–156	117	39–468	429
7.5%	39–156	117	39–468	429
10%	39–293	254	39–698†	659
Relative (3.194853778%)	39–98	59	293–468	175

Accuracy and calibration at the extended endpoint. Table 8 gives every added evaluation and the inherited boundary. From 429 to 698, SGCR's accuracy rises from 59.31% to 61.46%, while its ECE rises from 1.37% to 4.22%. Vanilla's corresponding accuracy rises from 58.67% to 61.51%, and ECE rises from 10.75% to 11.40%. At 698, SGCR also has lower Brier score (0.5003 versus 0.5253) and candidate NLL (0.9398 versus 0.9990). The observed accuracy difference is -0.048 percentage points for SGCR, without an equivalence or superiority claim. Extending training improves both methods' endpoint accuracy relative to step 429, but calibration does not improve monotonically.

Table 8: **Validation observations in the Qwen–MedMCQA continuation.** Accuracy and ECE-10 in percent, seed 1. Step 429 is the inherited boundary; the remaining seven rows are additional observations. Both methods use identical evaluation steps and denominators.

	Vanilla		SGCR	
Step	Accuracy ↑	ECE-10 ↓	Accuracy ↑	ECE-10 ↓
429	58.67	10.75	59.31	1.37
468	59.19	4.46	58.26	3.13
507	59.79	12.70	59.60	8.01
546	58.67	8.55	60.29	3.46
585	61.49	7.74	60.27	5.75
624	60.08	10.34	60.75	6.70
663	61.25	11.24	61.01	7.87
698	61.51	11.40	61.46	4.22

This pair was chosen for further training after inspecting the earlier results, and 698 was fixed as the endpoint before continuing. The comparison uses one seed and has no independent test confirmation. Additional checkpoints and repeated evaluations do not count as independent training runs. The original threshold grid, every qualifying interval, and full-precision observations are retained in the accompanying archive. These results do not establish a rule for stopping training or describe calibration beyond step 698.

E EXPERIMENTAL PROTOCOLS AND MEASUREMENT CONVENTIONS

E.1 TRAINING DATA, RESPONSE INTERFACE, AND OPTIMIZATION

The main MMLU experiments use 99,842 auxiliary-training examples, 1,531 official validation examples, and 14,042 official test examples. At evaluation, the model receives the question and its labeled choices, a request to return an answer in JSON, and the assistant prefix `{"answer": "`. The next-token logits provide candidate probabilities for A–D. The chat template disables thinking where supported. Candidate tokenization and the answer prediction position are audited. The answer-only training mask supervises the answer letter in the teacher-forced response while retaining the surrounding response text as context. Its NLL normalizes over the full vocabulary. Full-response variants instead supervise all designated assistant-response tokens.

Full-parameter training uses BF16, sequence-length limit 3,072, AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, gradient clipping at 1.0, and global batch size 256. In the main four-GPU configuration, microbatch size four and 16 accumulation steps preserve this global batch size. Learning rate peaks at 5×10^{-5} after 10% warmup and follows cosine decay. Standalone one-epoch scale runs use 390 planned updates and 39 warmup updates; three-epoch trajectories use 1,170 planned updates and 117 warmup updates. A step number alone does not specify the learning rate. Seed-1 scale experiments use A100 hardware and the additional seed-2/3 scale runs use A800 hardware under the same core recipe. Cross-seed variation consequently includes this hardware difference.

MedMCQA replication trains on 178,726 official-training examples after a frozen 4,096-example calibration-development split. The official labeled validation set contains 4,183 examples and is the primary outcome set; the unlabeled official test split is excluded. A three-epoch run has 2,094 updates (698 per epoch). Milestones are steps 0, 70, 175, 349, 524, 698, 1,047, 1,396, and 2,094. The development split is selected independently of outcomes. The validation set is used for mechanism evaluation across checkpoints, so it is not an untouched final confirmatory test set. This is a limitation of the cross-task evidence.

E.2 METRICS AND PROBABILITY SPACES

For n evaluation examples and bins $I_1, \ldots, I_B$ partitioning confidence into equal-width intervals,

$$\mathrm{ECE}_B = \sum_{b=1}^{B} \frac{|I_b|}{n} \left| \frac{1}{|I_b|} \sum_{i \in I_b} c_i - \frac{1}{|I_b|} \sum_{i \in I_b} \mathbf{1}[\hat{y}_i = y_i] \right|, \tag{21}$$

$$\mathrm{Brier} = \frac{1}{n} \sum_i \sum_{a \in \mathcal{A}(x_i)} \big(q_i(a) - \mathbf{1}[a = y_i]\big)^2, \qquad \mathrm{NLL}_{\mathcal{A}} = -\frac{1}{n} \sum_i \log q_i(y_i). \tag{22}$$

Empty bins contribute zero. Brier is summed over candidates, not averaged over the number of candidates. Candidate NLL is in natural-log units; the historical column name `nll_normalized_options` indicates probability renormalization over candidates, not division of NLL by $\log K$. The legacy format table additionally contains a separately normalized NLL field, which is not substituted for candidate NLL here. For a uniform four-choice predictor, Brier is 0.75 and NLL is $\log 4$; for ten choices they are 0.9 and $\log 10$.

Wrong confidence is $\mathbb{E}[c \mid \hat{y} \neq y]$ on the specified evaluation split. When evaluated at successive checkpoints, this conditions on changing sets of examples. It cannot by itself show that a fixed set of mistakes becomes more confident. The late-branch experiment therefore also tracks a fixed step-390 Wrong set. Similarly, W_t, C_t in the mechanism analysis denote *training* correctness; evaluation errors do not define the update groups.

Every ECE value, intervention effect, trajectory, and reliability plot in this paper uses $B = 10$. The intervals are $[b/10, (b+1)/10)$ for $b = 0, \ldots, 8$, and $[0.9, 1]$ for the last bin. Values are recomputed from saved predictions or taken from the original evaluation's recorded ten-bin field. Intervention changes subtract before and after ECE computed with these same edges. This common resolution is a retrospective reporting choice; the archived protocols and source outputs remain unchanged.

E.3 Statistical units and experimental reuse

The gradient and Wrong-fraction studies rerun the original model/seed training configurations separately to obtain the required checkpoints. Their validation evaluations are kept separate from the original suite's test curves. The runs that save checkpoints for same-example tests are also separate and are not pooled with these studies. The seed-1 experiment crossing output and loss-effective scale reuses its high/high and low/low conditions from the initial-scale study and trains the other two conditions. Seeds 2 and 3 add eight factorial runs. The continuous-scale analysis reuses three earlier conditions, optimizer studies include short diagnostic runs, and the full-response scale comparison reuses its two unmodified baselines. Reanalyses add measurements, not independent training replicates.

Across-seed standard deviations use the sample denominator $n - 1$ with $n = 3$. Gradient bands show this variation across seeds. Intervals for the Wrong-fraction experiment use 2,000 bootstrap resamples of 20 independently drawn training batches; intervals for the same-example experiment use paired early/late batches. These intervals describe batch variation at the chosen checkpoint and evaluation set. They do not estimate variation over future training runs. Confidence intervals are pointwise, without a multiple-comparison adjustment across checkpoints. The sequence of studies is exploratory. The fixed protocols for follow-up tests do not retroactively preregister the entire research program.

F Reliability Diagrams and Preserved Logit Measurements

The reliability diagrams and logit curves use 36 saved sets of test predictions: nine checkpoints for each of three Qwen Instruct seeds and one Base seed. Each contains the same 14,042 test-question IDs, gold answer indices, and four candidate logits. The 505,512 records include repeated predictions on these questions across checkpoints. Sample IDs and gold labels are checked for agreement before computing the curves. This reanalysis uses saved outputs and requires no model inference or retraining.

We recompute candidate probabilities with a numerically stable softmax, calculate ECE-10, and recover every state's archived accuracy, Brier score, mean confidence, and wrong confidence to numerical precision. In Figure 1, bars occupy the corresponding confidence bins and their heights give empirical accuracy. Hatching marks the gap between each bin's accuracy and its actual mean confidence, as defined in Equation 2. Histograms use the identical edges and report each bin's share of the entire test set. Correct and Wrong bars are stacked, so each complete histogram sums to 100%. The plots report benchmark frequencies, and all bin counts are retained in the analysis tables. The supplementary seed panels plot bin accuracy against mean confidence as descriptive ten-bin reliability curves.

NLL numerical convention. The archived CGB summaries evaluate $-\log \max(q_y, 10^{-12})$. At the original Instruct initialization, this floors 362 of 14,042 gold probabilities and gives NLL 4.4421. Recomputing $-\log q_y$ directly with log-softmax gives 4.5134. No gold probabilities reach that floor at subsequent saved Instruct checkpoints, so their two NLL definitions agree. The new proper-score figure uses the unfloored NLL. Historical tables retain their recorded NLL values and are identified as historical comparisons rather than silently substituted with the new calculation. This numerical convention affects an absolute initial NLL value; it does not change ECE, reliability curves, logit geometry, or the Wrong-fraction experiment's measured ECE effects.

Fixed question groups and token identities. For each seed, initially Wrong and Correct groups are defined at step zero. Their membership is then fixed. For each question, the distractor is the largest-logit incorrect candidate at step zero; this token is also held fixed. Figure 7c,d compares that token and the gold answer after subtracting the mean of the four logits separately at each checkpoint. It does not redefine the distractor using later outcomes. Means first aggregate questions within a seed; bands summarize variation across the three seed means. The fixed-group panels start at step 39 because initialization's much larger logits would obscure the later differences. The full initial-to-final norm trajectory is shown in panel (b), and all centered-logit means, medians, and quartiles are available in the analysis tables.

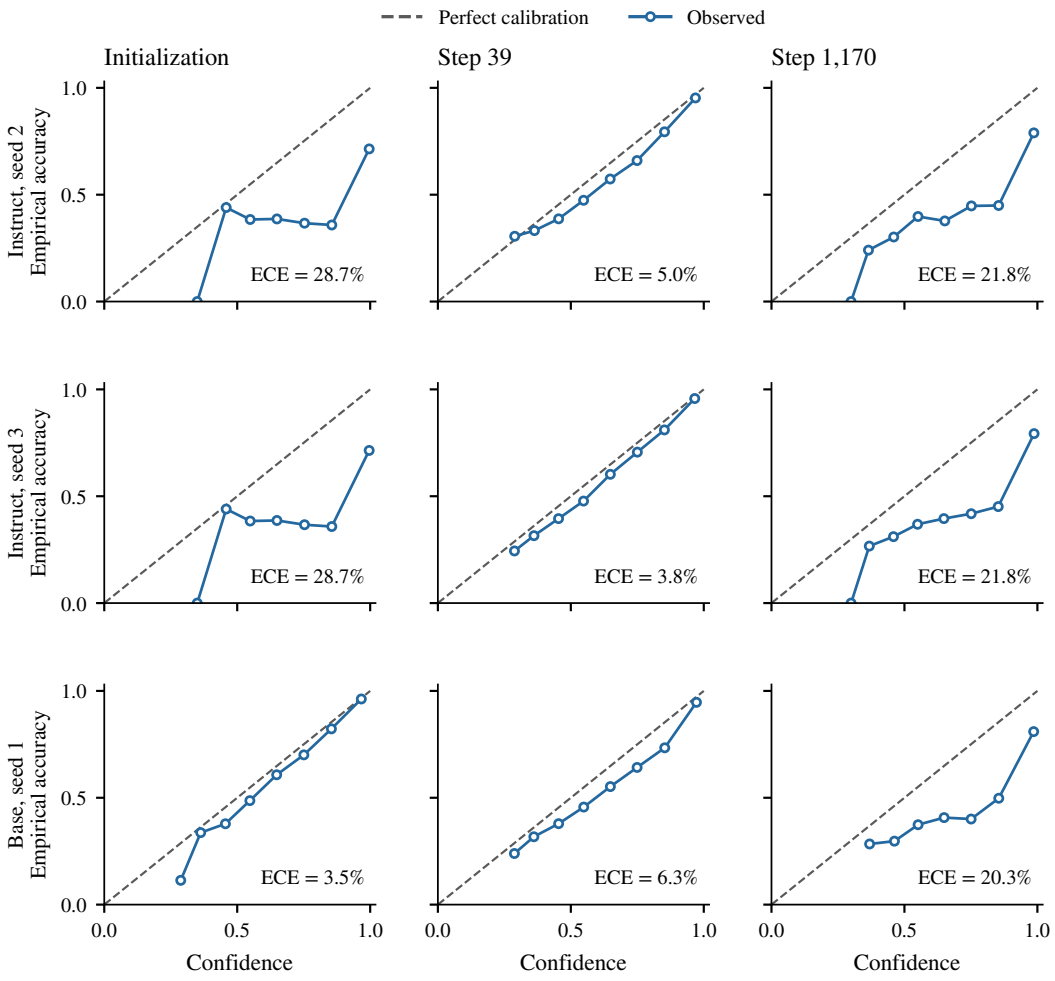


Figure 12: Reliability curves for the other two Qwen Instruct seeds and the Base control. Every panel uses 14,042 MMLU test questions, the same 10 equal-width confidence bins, and the same axis ranges. Rows identify the model and seed; columns identify fixed checkpoints. Base starts near the diagonal and becomes overconfident. These are individual training runs, with no pooling across seeds.

G Additional Trajectory Results and the Temperature Baseline

Figure 13 reports accuracy, Brier score, and candidate NLL throughout training, computed from saved logits. Figure 14 separately shows the independent runs in which early and late checkpoints were saved for later experiments.

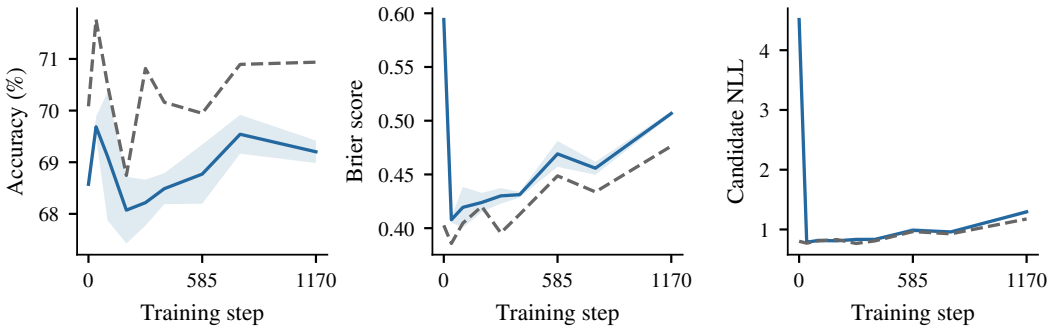


Figure 13: **Complementary trajectory measurements.** Accuracy, Brier score, and candidate NLL for the original MMLU test suite. Instruct lines show three-seed mean ± SD; Base is one run. NLL is recomputed with log-softmax without a probability floor.

Table 9: Original trajectory suite, MMLU test; ECE uses 10 bins. The minimum is the minimum among measured checkpoints for that run; its average is not the minimum of an averaged curve.

Model / seed	Initial ECE ↓	Min. ECE ↓	Final ECE ↓	Initial acc. ↑	Final acc. ↑
Instruct / 1	0.2874	0.0439	0.2171	0.6857	0.6895
Instruct / 2	0.2874	0.0429	0.2179	0.6857	0.6933
Instruct / 3	0.2874	0.0376	0.2183	0.6857	0.6933
Base / 1	0.0354	0.0354	0.2028	0.7008	0.7094

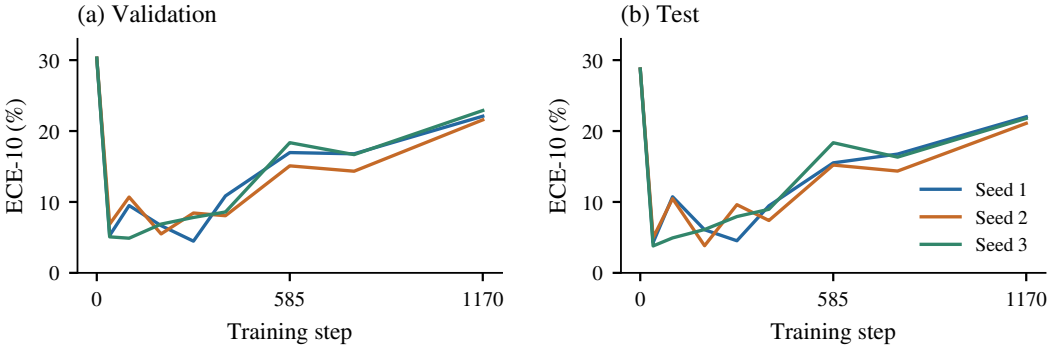


Figure 14: Independent Qwen runs with checkpoints saved for later experiments. Each line is one seed. These validation and test ECE-10 curves come from runs separate from the main gradient and Wrong-fraction experiments; they are not counted as additional seeds for those experiments.

The earlier single-seed trajectory study reaches Instruct ECE 0.0609 at step 195 and 0.2169 at step 1,170. This historical replication differs from the later suite and is not appended as an extra seed to its summary. The turning-point comparison uses the saved validation trajectory from the study of gradients at successive checkpoints. This evaluation pass retains ECE-10 for every checkpoint; it is distinct from the gradient pass and from the original suite's test evaluation.

The separately saved runs reach minimum validation ECE-10 values 0.0447, 0.0549, and 0.0489, and end at 0.2209, 0.2158, and 0.2289. They also show an early decrease followed by a later increase in calibration error. The gradient analysis uses separately reconstructed checkpoints.

Table 10: Historical post-hoc calibration in the separate rank–scale reanalysis; NLL retains its recorded probability-floor convention. All outcomes are MMLU test; temperatures are fitted on validation. SFT step 390 belongs to a 1,170-step schedule.

Condition	Accuracy ↑	ECE-10 ↓	Brier ↓	Candidate NLL ↓
Untrained, raw	.6857	.2875	.5942	4.4419
Untrained, temperature	.6857	.0636	.4256	.8240
Untrained, temperature + bias	.6843	.0598	.4254	.8209
SFT step 390, raw	.6887	.1109	.4350	.8438

The fitted scalar temperature is 10.7123. Decomposing the difference between the two checkpoints assigns 90.4% of the ECE change, 89.9% of the Brier change, and 92.4% of the NLL change to scale. These percentages describe the saved logits; they do not measure causal shares of SFT learning. Initial and step-390 predictions agree on only 74.0% of test examples, so the large scale term does not mean that predicted answers stay unchanged during training.

H Rescaling Logits at Initialization and in the Loss

H.1 Scale interventions and results across seeds

The initial-scale multiplier is the ratio of validation-fitted temperatures for the original Instruct and Base models: $s = 10.6957200433/1.2638376063 = 8.4628911104$, with reciprocal approximately 0.118163. The fit minimizes candidate NLL on the 1,531 official validation questions using candidate-only FP32 unembedding; it does not select a scale from the test calibration curve. The implementation multiplies the final RMSNorm weight once at initialization. The embedding matrix and tied language-model head are left unchanged. In ideal arithmetic a positive scale preserves the initial argmax; the large-scale Base manipulation has 99.843% agreement on the complete MMLU test evaluation. This finite-precision discrepancy qualifies claims of exact prediction preservation.

For the Instruct factorial experiment, high raw corresponds to the original initial scale and low raw to $1/s$. Conditions high/high, high/low, low/low, and low/high use fixed loss temperatures $1, s, 1, 1/s$, respectively. Thus high/low and low/low start with matched effective logits, while high/high and low/high start with matched effective logits. Temperature is constant within each such run. "High" and "low" refer to the controlled initial scales, not fixed values of the logits throughout training.

Table 11: MMLU test endpoints at 390 updates for the raw/effective scale factorial experiment. Mean ± sample SD over three training seeds.

Raw / effective	Accuracy ↑	ECE-10 ↓	Brier ↓	NLL ↓
High / high	0.7023 ± 0.0038	0.0486 ± 0.0067	0.3975 ± 0.0036	0.7580 ± 0.0068
High / low	0.6940 ± 0.0035	0.2678 ± 0.0040	0.5604 ± 0.0073	3.5245 ± 0.1576
Low / low	0.6952 ± 0.0012	0.0501 ± 0.0102	0.4079 ± 0.0032	0.7800 ± 0.0078
Low / high	0.7015 ± 0.0023	0.3722 ± 0.0017	0.6312 ± 0.0026	1.1709 ± 0.0043

The endpoint table uses all three training seeds without pooling item-bootstrap samples. Mean confidence is approximately 0.962 in high/low but 0.329 in low/high, against approximately 0.694 and 0.702 accuracy, respectively. A high ECE in the latter condition is therefore due to underconfidence. Calibration error is not synonymous with overconfidence.

For seed 1, pairs with the same loss-effective scale have cosine similarities 0.99969 and 0.95840 between their logit changes. Their mean is 0.9790, whereas pairs with the same output scale average -0.1162. A separate AdamW control starts with zero moments and a nonzero learning rate. After one update, pairs with the same loss-effective scale have similarities 0.4837 and 0.5854. This control avoids treating the original warmup iteration, whose learning rate was zero, as evidence about update direction. It also shows that similar final logit changes do not imply identical first parameter updates.

Initial-scale effects also transfer to the held-out MedMCQA evaluation in the one-epoch MMLU-trained crossover: ECE changes are -0.2876, $+0.0026$, $+0.0461$, and -0.2471 for original Instruct,

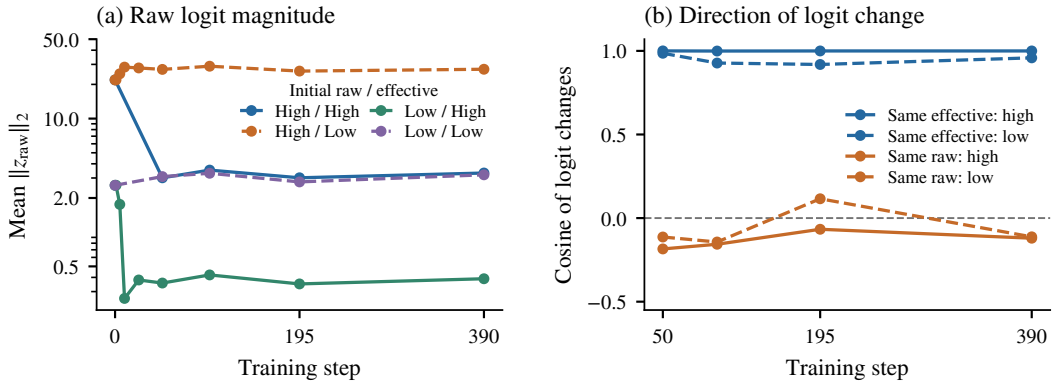


Figure 15: **Matching the scale in the loss produces similar logit changes.** Seed-1 experiment crossing output and loss-effective scale. (a) Raw centered-logit norm, with initial raw/effective scale indicated. (b) Cosine between flattened logit changes $z_t - z_0$ on the same test questions: pairs match either effective or raw scale. Every measured comparison is shown. Logit changes use one seed; Table 2 reports final results over three seeds.

low-scale Instruct, original Base, and high-scale Base. These are evaluation transfers from a single training seed, distinct from the later MedMCQA-trained three-seed replication.

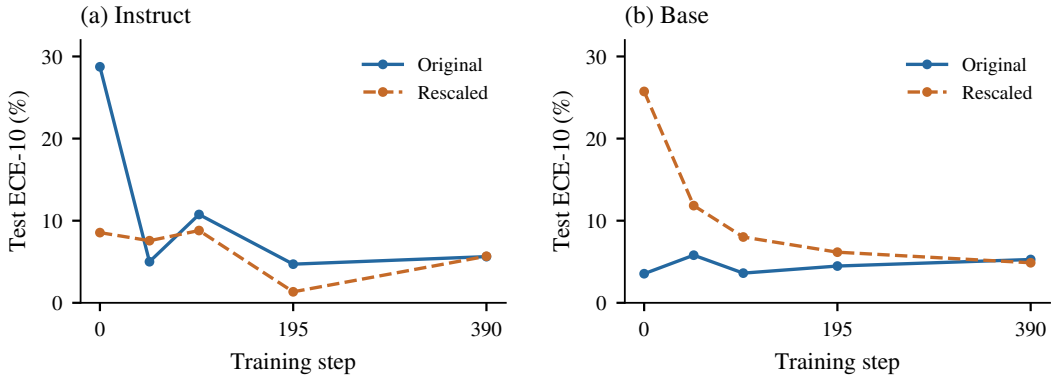


Figure 16: **Rescaling before training changes whether SFT improves calibration.** Complete crossover trajectories, one seed per condition. Instruct is rescaled down by $s = 8.4629$; Base is rescaled up by the same factor. All runs use the separate 390-update schedule and report test ECE-10.

H.2 What residual size reveals about each training group

For full-vocabulary logits $\ell_{\text{raw}} = gf_\theta$, Equation 6 gives

$$\nabla_\theta \mathcal{L}_\tau = gJ_\theta^\top \left[\text{softmax}(gf_\theta/\tau) - e_y\right], \tag{23}$$

when g is treated as a fixed output multiplier in this coordinate description. The ratio g/τ selects the effective distribution, while g additionally multiplies the parameter gradient. Our actual intervention is implemented through final-normalization parameters; this coordinate description does not assert that every trainable parameter transforms identically under it. The controlled residual and trajectory measurements are therefore necessary.

For a group G, define its share of summed residual norms as $M_G = \sum_{i\in G}\|p_{\text{eff},i} - e_{y_i}\|_2 / \sum_i \|p_{\text{eff},i} - e_{y_i}\|_2$. On the 4,000-example initial sample, Wrong examples account for 96.45% at high loss-effective scale, while Correct examples account for 85.31% at low loss-effective scale. Scale therefore changes which examples supply most of the output residual. To determine their effect on the parameter update, we also need the Jacobian. Matching these aggregate shares does not guarantee the same effect for each example.

I Additional Scales and Optimizers

A single-seed continuous sweep extends the discrete factorial control. Joint-scale conditions cover $g = \tau \in \{0.5, 1, 2, 4, 8.4629\}$, together with additional effective-temperature conditions. The study comprises six new runs and three reused conditions. In the joint-scale family, endpoint accuracy spans 0.00577, minimum pairwise prediction agreement is 0.9276, and mean cosine similarity of logit changes is 0.9719. Fitted raw temperature and centered-logit norm scale nearly linearly with g ($R^2 = 0.999964$ and 0.999965). Coefficients of variation of T/g and norm$/g$ are 0.0066 and 0.0054. These observations support approximate equivalence under this recipe, not identity of predictions or gradients.

The optimizer control isolates the global gradient multiplier with 50 constant-learning-rate updates and no warmup. The scale-8/scale-1 gradient-norm ratio is 8.0004, while the output residual cosine is 1. The endpoint comparisons below show why both cosine and prediction agreement should be reported.

Table 12: Optimizer controls for scale 1 versus 8. Each row is a pair of short runs; these are not extra three-epoch training seeds.

Update rule	Cosine similarity	Top-1 agreement
AdamW, clipping	.9998	.8760
AdamW, no clipping	.9999	.9540
SGD, same learning rate	.9645	.2510
SGD, learning rate divided by scale	1.0000	.9875
SGD, clipping active	.9986	.9935

Adaptive normalization can approximately remove a common gradient multiplier, and active clipping can constrain its norm. For plain SGD, changing the multiplier changes the step size unless the learning rate compensates. These properties help explain why some pairs of runs are similar. A large cosine similarity or a nearly linear relation between output scales does not establish equivalent learning under every optimizer.

J Gradient Measurements and Changes in the Wrong Fraction

J.1 A smooth calibration objective for gradient analysis

Let $a_i = \mathbf{1}[\hat{y}_i = y_i]$ at the checkpoint and $c_i = q_i(\hat{y}_i)$. Predictions and correctness indicators are fixed during differentiation. With centers $\mu_b = (b - 1/2)/B$ and smoothness parameter κ, define

$$w_{ib} = \frac{\exp[-(c_i - \mu_b)^2/\kappa]}{\sum_k \exp[-(c_i - \mu_k)^2/\kappa]}, \quad W_b = \sum_i w_{ib}, \quad \bar{c}_b = \frac{\sum_i w_{ib} c_i}{W_b}, \quad \bar{a}_b = \frac{\sum_i w_{ib} a_i}{W_b}. \tag{24}$$

The smooth objective is $\mathcal{R} = n^{-1} \sum_b W_b (\bar{c}_b - \bar{a}_b)^2$. Its gradient includes how confidence changes the bin weights. All reported projections, including those for the same examples at early and late checkpoints, use $B = 10, \kappa = 0.05$. These settings come from the archived, prespecified sensitivity grid. Holding $B = 10$, all 108 measurements across 36 checkpoints and $\kappa \in \{0.03, 0.05, 0.08\}$ give positive Wrong and negative Correct projections. Ten-bin projections for individual data shards were not saved. We therefore report aggregate point estimates and across-seed SD, without claiming significance from a shard bootstrap.

Proposition 3 applies at a state where the surrogate is differentiable and its gradient is Lipschitz on the update segment. For an update with a fixed preconditioner P_t, its linear term becomes $-\eta \langle h, P_t g_t \rangle$. A native AdamW update additionally depends on previous moments and decoupled weight decay; its group effect is not determined by the Euclidean projection alone. Group NLL gradients exclude weight decay. A ten-bin projection of the archived virtual AdamW update is unavailable and is not included in the quantitative comparison.

Group gradients are computed across all trainable parameters from a fixed 32,768-example training pool, partitioned by correctness at each checkpoint. The calibration gradient uses the official validation

set. Full-training-set proportions weight these estimates. Reconstructing a mixed gradient from estimated group gradients introduces a numerical and sampling discrepancy. Its relative residual has median 0.0454, 90th percentile 0.0766, and maximum 0.2858. The largest residual is material and constrains precise quantitative interpretation. Update interventions provide a separate check of the group-level interpretation.

Table 13: Changes in the factors that determine the Wrong group's weighted projection, from step 0 to 1,170, Qwen Instruct. Percentages are declines within each run from initialization to the endpoint. They are neither causal attribution percentages nor evidence of a monotonic decrease. Figure 4 shows the complete curves.

Seed	$\pi_W \rho_W$: initial $\rightarrow$ final	Wrong fraction	Gradient norm	Alignment
1	$17.7603 \rightarrow 0.04246$	96.35%	86.01%	53.18%
2	$17.7614 \rightarrow 0.03432$	96.51%	92.82%	22.91%
3	$17.7605 \rightarrow 0.03789$	96.76%	91.56%	21.99%

The complete frequency, norm, alignment, and weighted-projection curves are shown together in Figure 4 in Section 4.2.

J.2 How much does the Wrong fraction need to increase?

For each state, the requested Wrong fractions $0, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50, 0.75, 1$ correspond to actual Wrong counts $0, 5, 13, 26, 51, 77, 128, 192, 256$ in a batch of 256. Sampling is without replacement within a batch. Native model parameters, optimizer moments, optimizer step count, scheduler state, and the checkpoint learning rate are restored before each intervention. The outcome is one update later; updates are not chained. A natural-random condition samples from the original training distribution. Twenty batches per condition are evaluated against the same held-out validation examples.

For an idealized SGD step with fixed group gradients, the predicted critical fraction is $\alpha_{\text{grad}} = -\rho_C/(\rho_W - \rho_C)$. The empirical critical fraction is obtained by linear interpolation between adjacent harmful and beneficial *mean* effects, without extrapolation. This mean crossing differs from the minimum tested fraction whose entire confidence interval is beneficial. We report strict natural-harm-to-controlled-benefit separately from the existence of a crossing.

Section 5.1 and Figure 5 report the batch-composition effects. Figure 17 retains these curves alongside the gradient-threshold and timing comparisons.

Table 14: Wrong fractions at which the update changes from harmful to beneficial, at all 12 checkpoints. Natural fractions use correctness over the full training set; empirical crossings use mean update effects. A strict flip requires the natural-update confidence interval to be harmful and a controlled-update interval to be beneficial.

Seed	Step	Natural π_W	Gradient threshold	Empirical crossing	Strict flip
1	195	0.0893	0.1113	0.1780	Yes
1	293	0.0559	0.0929	0.2147	Yes
1	390	0.0348	0.1039	0.1199	Yes
1	585	0.0204	0.0154	0.0295	Yes
2	195	0.0798	0.1373	0.1855	Yes
2	293	0.0571	0.0559	0.1232	Yes
2	390	0.0360	0.0872	0.0419	No
2	585	0.0191	0.0190	0.0362	Yes
3	195	0.0768	0.1033	0.1594	Yes
3	293	0.0580	0.0513	0.0264	No
3	390	0.0350	0.0274	0.0902	Yes
3	585	0.0188	0.0132	0.0401	Yes

All 12 states have a mean crossing and a clearly beneficial controlled condition. Ten have a strictly harmful natural baseline as well. Eleven natural fractions fall below the empirical crossing, but

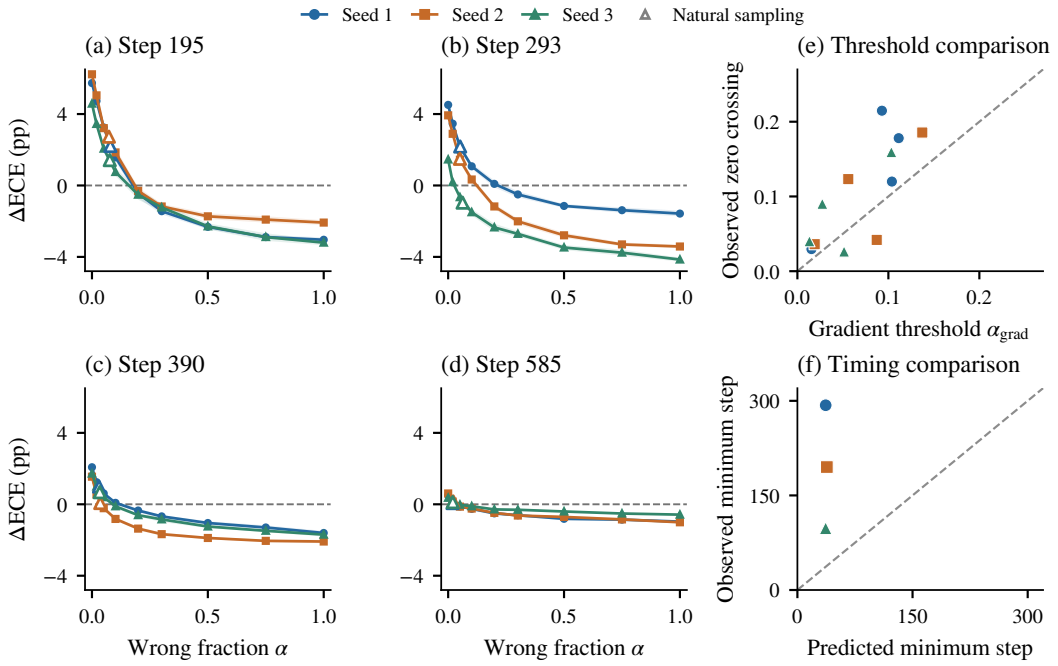


Figure 17: **Increasing the Wrong fraction can improve one update, but gradient predictions have limits.** (a–d) One-step validation ECE-10 change versus Wrong fraction, with the native AdamW state restored before each update. Lines are seed means over 20 batches; bands are pointwise 95% batch-bootstrap intervals; open triangles denote natural sampling. (e) Thresholds predicted from SGD and the smooth calibration objective versus measured crossings, using separate runs matched by model, seed, and step. (f) First crossing of the weighted gradient balance versus the minimum of the gradient study's saved validation curve. Every tested checkpoint is shown.

not all of these natural effects are statistically separated from zero. The mean absolute discrepancy between the gradient threshold and empirical crossing is 0.04729. Relative error uses the gradient threshold as denominator and has median 75.05%; the denominator choice matters near zero. The gradient and intervention states are separate reconstructions matched by model, seed, and step. Their comparison includes reconstruction variability and cannot estimate the approximation error at an identical parameter state. The maximum absolute accuracy change among the reported condition means is 0.01398. This bounded movement helps characterize the intervention but does not imply exactly fixed predictions.

J.3 Paired prediction changes after a group update

Figure 18 extends Figure 3b to all three seeds and four checkpoints, including natural sampling. We recover the original four candidate logits and probabilities for 720 archived updates: 20 batches for each of three conditions at 12 states. Sample IDs, gold answers, and source hashes align with the previously preserved confidence/correctness arrays. Recomputed ECE-10 effects and archived mean entropy changes agree to numerical precision. No new model training or inference is involved.

For a state t, define the fixed evaluation group $W_t^{\text{val}} = \{x : k_t(x) \neq y_x\}$. We report

$$\Delta p_{\text{old,W}} = \frac{1}{|W_t^{\text{val}}|} \sum_{x \in W_t^{\text{val}}} \left[q_{t+1,k_t(x)}(x) - q_{t,k_t(x)}(x) \right]. \tag{25}$$

Both the group and the tracked answer are defined before the update. The post-update probability is taken at the original answer even when it is no longer the winner. This differs from subtracting maximum probabilities or comparing the changing sets of errors at t and $t+1$. Entropy $H(q) = -\sum_a q_a \log q_a$ uses natural logarithms; its mean change and the mean change in centered-logit norm

use all 1,531 validation questions. The norm uses $z = HS_x\ell_\theta(x)$ from Section 2. The before/after predictions are paired by sample ID.

Every empirical point in Figure 3c and Figure 18 is a mean over 20 independently sampled training batches at one restored state. The main prediction panel additionally shows all 20 trials at seed 1, step 390, the first seed of the existing worked checkpoint. Pointwise 95% percentile intervals use 10,000 bootstrap resamples of the batch trials, conditional on the checkpoint and evaluation set. They do not treat questions or different checkpoints as independent training replicates. The four-option probabilities in Figure 3a are illustrative; their arrows show $e_y - p$ for a directly optimized four-class logit vector. Actual held-out changes come from full-vocabulary NLL parameter updates through the shared network and optimizer.

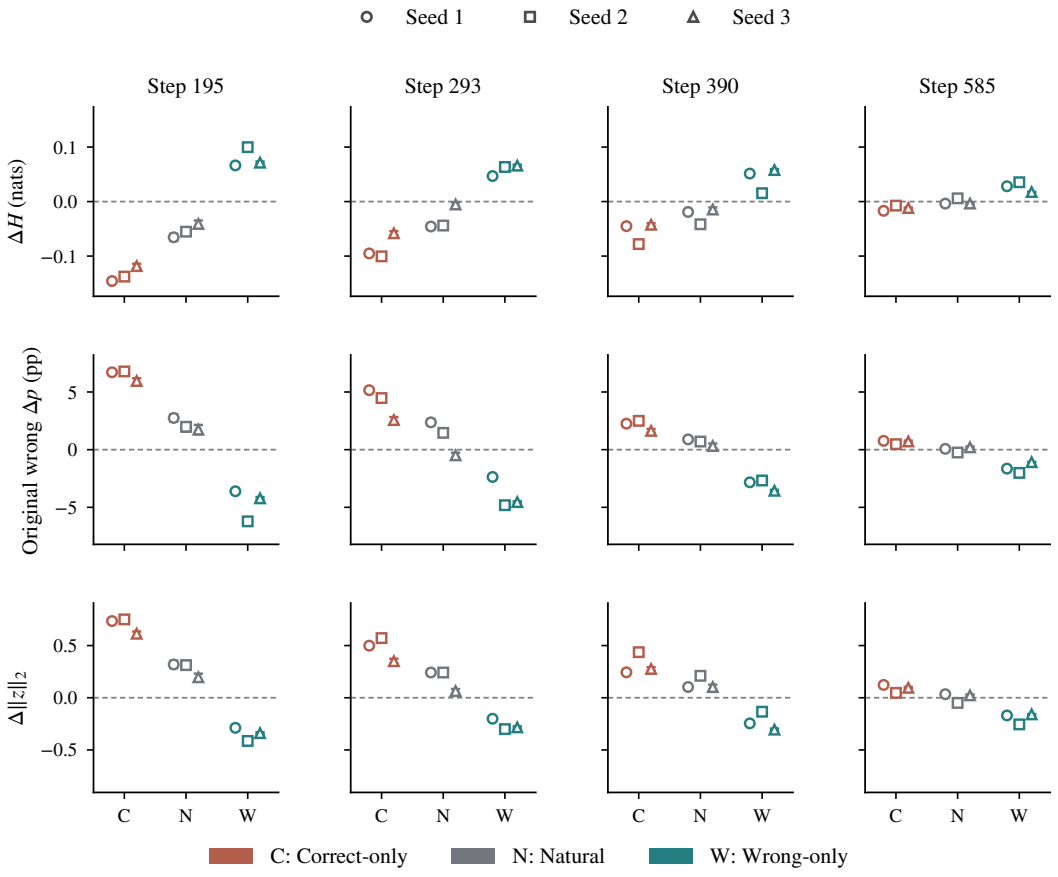


Figure 18: **Prediction changes after updates from each training group.** Rows show entropy change, probability change of the original wrong answer on questions answered incorrectly before the update, and centered-logit norm change. Columns identify the checkpoint. C, N, and W denote Correct-only, natural, and Wrong-only training batches. Shapes identify seeds; points and intervals are batch means and 95% batch-bootstrap intervals. Every panel uses the same held-out validation questions; the incorrectly answered subset is defined before each update.

K Updating on the Same Examples Early and Late in Training

This Qwen experiment saves the actual early and late checkpoints for later comparison, rather than reconstructing deleted checkpoints. The chosen pair is steps 98 and 390 for all three seeds. WC, WW, CC, and CW indicate whether the examples are Wrong or Correct at each checkpoint, in that order. WC pool sizes are 8,027, 7,775, and 7,478. Membership depends on training correctness at both checkpoints, so these groups are not random samples of all training examples.

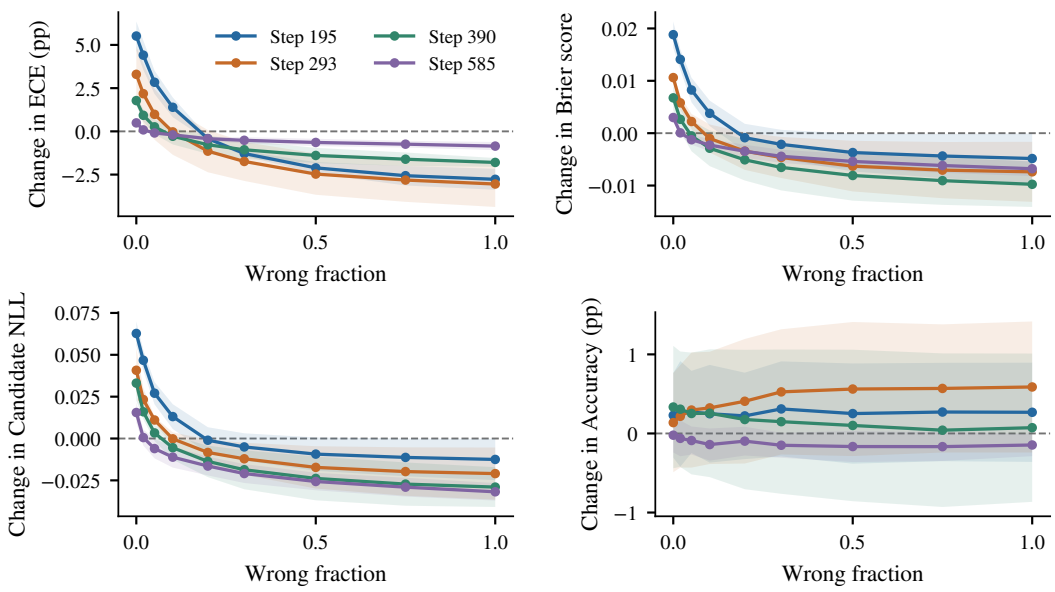


Figure 19: Accuracy, Brier score, and NLL after changing the Wrong fraction. Each line fixes a checkpoint and averages three seed-level effects; bands are sample SD across seeds, not batch-bootstrap intervals. Every tested fraction is shown. Accuracy can change, especially at earlier checkpoints; at step 585, the mean accuracy response is slightly negative.

For the primary test, a single fresh SGD update is applied to FP32 parameters with BF16 forward/backward computation, learning rate 10^{-5}, batch size 256, no momentum, and no weight decay. The learning rate was chosen on a non-primary numerical-stability probe and then locked across seeds. Twenty paired batches per group use identical examples at early and late states. The optimizer reset makes the intervention comparable across states without importing different native momentum histories. Gradient projections use the same paired batches. Ten paired batches per group provide a separate native-AdamW sensitivity analysis, which does not replace the primary SGD result.

For each paired batch, measure the difference between its late and early ECE effects as $\mathrm{RFE} = \Delta\,\mathrm{ECE}_{\mathrm{late}} - \Delta\,\mathrm{ECE}_{\mathrm{early}}$. A strict WC sign change requires the early confidence interval to lie below zero, the late interval above zero, and the paired RFE interval above zero. We assess the control groups and gradient criteria separately. Table 15 reports all primary WC intervals, including the non-significant late effect for seed 1.

Table 15: Qwen updates on the same examples at saved early and late checkpoints. All entries are ECE-10 change $\times 10^3$; lower is better for the Early/Late update effects. RFE is their late-minus-early contrast. Intervals are 95% paired-bootstrap intervals over 20 batches within one training seed.

Seed	Effect	Mean	95% CI
1	Early	-1.7215	[-1.9961, -1.4198]
1	Late	+0.0132	[-0.0521, +0.0475]
1	Rfe	+1.7348	[+1.4638, +2.0082]
2	Early	-1.1970	[-1.3670, -1.0489]
2	Late	+0.1689	[+0.0703, +0.2991]
2	Rfe	+1.3659	[+1.1907, +1.5522]
3	Early	-2.8300	[-3.2577, -2.4179]
3	Late	+0.0416	[+0.0356, +0.0479]
3	Rfe	+2.8716	[+2.4808, +3.2861]

The mean RFE is 0.001991 ± 0.000785. The mean projected WC gradient changes from 22.49 ± 4.63 to -1.360 ± 0.125. These estimates describe the chosen examples at the two checkpoints. Their effect is much larger early than late, and the late ECE change is small in absolute terms. Thus, the

same examples can affect calibration differently as the model learns. The experiment does not isolate becoming Correct as the cause: selecting examples that change from Wrong to Correct also selects examples whose margins and representations have changed.

L Cross-Model and Cross-Task Replication

The Llama–MMLU replication uses three training seeds and ECE-10 on official validation as primary. Its mean initial ECE is 0.1680 ± 0.0014, and the mean change to the measured trajectory minimum is -0.0932 ± 0.0113. We compare the magnitudes of the accuracy and ECE changes using the frozen criterion $|\Delta\text{accuracy}|/|\Delta\,\text{ECE}| \leq 0.1$. No seed passes; the mean ratio is 0.530 ± 0.349.

Wrong-only updates give mean $\Delta\,\text{ECE} = -0.03790 \pm 0.00416$ across the three seed point estimates, while Correct-only updates give $+0.04436 \pm 0.01065$. The Wrong fraction decreases from 0.1364 ± 0.0022 to 0.0324 ± 0.0130 between the chosen early and late checkpoints. On the coarse grid, the ECE change decreases with Wrong fraction, with mean Spearman correlation -1 in all seeds. The recorded aggregate criterion combines improvement from increasing the Wrong fraction with evidence that the update changes from harmful to beneficial. It does not provide a separate strict sign-change count comparable to MedMCQA, so Table 1 reports effect sizes.

For Llama WC updates, mean early ECE change is -0.000315 ± 0.001027, late change is $+0.0000155 \pm 0.0000390$, and the paired difference is $+0.000331 \pm 0.001005$. The strict same-example criterion fails in all three seeds, although the projected gradient changes from positive to negative in two. Figure 20 displays the individual seed effects on common axes.

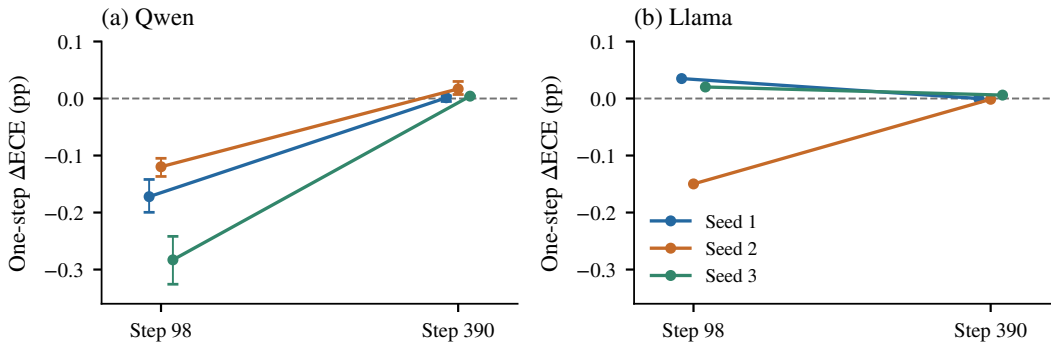


Figure 20: Same-example effects across models. Lines connect each seed's early and late mean effects. Qwen intervals are 95% paired-bootstrap intervals; Llama shows available seed point estimates. Both panels share the same vertical range.

Table 16: MedMCQA replication: mean ± sample SD of three seed point estimates per model. ECE is measured on official validation with 10 bins. Group effects summarize selected checkpoints within each seed, not every individual update.

Measurement	Qwen3-4B	Llama-3.2-3B
Initial-to-minimum Δ ECE ↓	$-.4093 \pm .0074$	$-.1240 \pm .0164$
Wrong-only Δ ECE ↓	$-.0101 \pm .0116$	$-.0317 \pm .0135$
Correct-only Δ ECE ↓	$+.0414 \pm .0057$	$+.0740 \pm .0341$
Natural-batch Δ ECE ↓	$+.0054 \pm .0080$	$+.0242 \pm .0196$
Fractional decrease in Wrong examples	$.8290 \pm .0038$	$.7579 \pm .0059$
Mean coarse-response Spearman	$-.9524 \pm .0595$	$-1.0000 \pm .0000$
Strict composition-flip seeds	2/3	3/3

The MedMCQA group-only intervention uses exactly restored native AdamW state, one update, batch size 256, and 20 batches per condition and selected checkpoint. Its test of Wrong fractions on a coarse grid uses ten replicates per condition and fractions 0, the natural Wrong fraction, 0.05, 0.10, 0.25, 0.50, and 1, truncated to the maximum feasible Wrong count when necessary. The frozen

selection rule favors the earliest two post-minimum group-probe checkpoints with a positive natural-update ECE change, with fallback checkpoints at steps 698 and 1,047. Consequently, the reported composition-flip rates are conditional on a selection designed to probe late harmful states; they are not prevalence estimates over all training checkpoints. The primary MedMCQA study measures the decrease in the Wrong fraction and does not execute the full parameter-gradient decomposition.

Both models pass the fixed seed-level criteria for early calibration improvement, opposite group effects, a decreasing Wrong fraction, and improvement from increasing that fraction. Individual checkpoints can still differ: Wrong-only updates improve calibration at 3, 2, and 2 tested Qwen checkpoints in seeds 1, 2, and 3. The stronger criterion requiring a strict change from harmful to beneficial passes in two Qwen seeds. Same-example WC interventions were not run in this cross-task study. Absolute ECE is not normalized across tasks or combined into one pooled effect.

M Task-Matched and Generic Full-Response SFT

The matched-format experiment uses 3,335 training and 334 validation examples for each of four conditions: MMLU answer-label-only, MMLU full-response, Alpaca full-response (Taori et al., 2023), and OpenAssistant full-response (Köpf et al., 2023). Strict English filtering leaves 3,669 eligible OpenAssistant examples, which sets the matched budget. Three epochs correspond to 39 updates, not 1,170. MMLU label-only and full-response conditions share the same response text and differ in the supervision mask. Full-response losses are averaged over supervised tokens. Example exposure is matched; token exposure and response-length distributions are not.

Table 17: MMLU test endpoints after matched-example training, one seed per condition. The full-response losses supervise all designated assistant tokens; label-only supervises the answer letter.

Model	Training condition	Accuracy ↑	ECE-10 ↓	Brier ↓	NLL ↓
Instruct	Alpaca-Full	0.6836	0.1336	0.4520	0.9514
Instruct	MMLU-Full	0.6746	0.1874	0.4956	1.1120
Instruct	MMLU-Label	0.6765	0.1987	0.5007	1.1422
Instruct	OA-Full	0.6857	0.1206	0.4470	0.9301
Base	Alpaca-Full	0.6952	0.0993	0.4259	0.9319
Base	MMLU-Full	0.6884	0.1714	0.4781	1.0253
Base	MMLU-Label	0.7035	0.1444	0.4340	0.9145
Base	OA-Full	0.7052	0.1064	0.4126	0.8480

All four Instruct conditions improve ECE relative to its high-confidence initialization. Base ECE increases in all four, but the full-response MMLU condition loses enough accuracy to fail the experiment's clean-degradation criterion. The comparison between Alpaca and OpenAssistant cannot attribute differences solely to synthetic versus human origin, because content and token distributions also differ. Five-shot evaluations are retained as a separate protocol view and are not mixed with the main zero-shot endpoint table.

Table 18: OpenAssistant full-response scale crossover: one seed, 39 updates. Original-scale conditions are reused from the matched-format experiment. ECE uses 10 bins.

Initial state	Initial acc. ↑	Final acc. ↑	Initial ECE ↓	Final ECE ↓
Instruct, original	0.6857	0.6857	0.2874	0.1206
Instruct, low scale	0.6857	0.6598	0.0855	0.2160
Base, original	0.7008	0.7052	0.0354	0.1064
Base, high scale	0.7008	0.6822	0.2574	0.1004

Brier changes in these four conditions are -0.1472, $+0.1007$, $+0.0099$, and -0.1096, and candidate NLL changes are -3.5120, $+0.3083$, $+0.0451$, and -2.6348. Thus the initial-scale dependence is visible beyond ECE. Both manipulated runs lose accuracy, preventing a claim of cost-free calibration repair. The experiments do not evaluate calibration of the generated full responses themselves.

N Predicting Calibration Changes and Controlling Longer Training Runs

N.1 The gradient balance does not identify when to stop training

The score computed at the current checkpoint uses a zero threshold and no future checkpoint information. Outcomes here use ECE-10 from the gradient study's associated saved validation trajectories, evaluated in a separate pass from the calibration-gradient computation. Across 24 Qwen Instruct intervals, sign accuracy is 0.833, balanced accuracy 0.832, and AUROC 0.825 for the next validation-ECE direction. Spearman correlation with the negative ECE slope is 0.677. Pearson correlation is 0.990 but is dominated by the large initial transition; a post-hoc analysis excluding step-zero intervals lowers Pearson correlation to 0.623 while retaining 0.810 sign accuracy. These adjacent intervals share training runs and are not independent model replications.

Table 19: First interpolated balance zero crossing versus the minimum of the associated saved validation trajectory. These are retrospective diagnostic comparisons, not an evaluated deployment selector.

Run	Observed minimum step	Predicted crossing	Absolute step error ↓
Instruct seed 1	293	36.83	256.17
Instruct seed 2	195	38.38	156.62
Instruct seed 3	98	36.80	61.20
Base seed 1	293	185.12	107.88

For Base, sign accuracy is 0.625, balanced accuracy is 0.357, and AUROC is 0.286 across eight intervals. The projection is more consistent with the initial direction of change than with the full trajectory. It therefore cannot be treated as a reliable predictor of hard-binned ECE throughout training.

N.2 Controlling scale and continuing from the same checkpoint

The long-horizon control study trains two initial states under standard NLL, a fixed effective-scale control, and a dynamic effective-scale control for 1,170 updates, giving six runs with one training seed per condition. Dynamic control reduces coarse residual-distribution mismatch to 0.0559, compared with 0.1719 under standard training and 0.0928 under fixed control. However, sample-level residual L_1 discrepancy is 0.7051 for dynamic versus 0.2392 for fixed control. Matching a coarse distribution does not preserve sample-level learning.

Table 20: Step-1,170 MMLU test outcomes after validation-fitted temperature scaling. Means are over two initial states with one training seed each, not two independent seeds. AURC is recalculated in this probability view.

Training rule	Accuracy ↑	Brier ↓	NLL ↓	AURC ↓
Standard	0.6874	0.4264	0.8198	0.1341
Fixed	0.6877	0.4313	0.8328	0.1315
Dynamic	0.6856	0.4423	0.8623	0.1388

The proper-score comparison in Table 20 uses a temperature fitted separately on validation at the corresponding checkpoint. Standard training has lower final post-hoc Brier and NLL than either controller in this study. On MedMCQA evaluation, standard post-hoc Brier/NLL/AURC are 0.5955/1.1066/0.2863 versus dynamic control's 0.6191/1.1549/0.2929. The controllers are therefore informative mechanism probes without demonstrated general post-hoc performance gains.

A separate late-branch experiment starts continuations from exactly matched step-390 parameters, optimizer state, scheduler, random-number state, and data position. On the fixed set of examples wrong at step 390, confidence grows by 0.1789 under standard continuation and 0.2249 under the one-sided controller. The controller fails to suppress this outcome. Its final post-hoc Brier/NLL/AURC (0.4243/0.8146/0.1297) slightly improve on standard continuation (0.4266/0.8194/0.1340), but a

fixed step-390 comparator gives 0.4185/0.7925/0.1288. This is a comparison with a specified earlier checkpoint, not a validated optimum obtained by tuning an early-stopping rule.

A scalar temperature preserves within-example argmax but can change the ordering of maximum confidence across examples in multiclass problems. For example, logits $(3, 0, 0, 0)$ and $(1, 0, -10, -10)$ have maximum confidences approximately 0.870 and 0.731 at temperature one, but 0.310 and 0.389 at temperature ten. Their cross-example order reverses. We therefore compute selective-risk rankings separately for each calibration view and never substitute raw AURC for temperature-corrected AURC.

O Earlier Experiments on Loss Functions, Response Formats, and Answer Labels

O.1 Loss-function comparisons from the initial full report

The historical objective sweep covers Qwen3-4B Instruct/Base and Qwen2.5-3B Instruct/Base (Yang et al., 2024). It comprises nine trained objectives per model plus one untrained baseline: 36 trained variants and four baselines, evaluated on MMLU validation and test. These are predominantly single-seed comparisons. Table 21 preserves all 40 test states. It supplies historical controls rather than additional standard-NLL trajectory replications.

Let p be the full-vocabulary target-token probability. The trained objectives are NLL $-\log p$, $1 - p$, $(1 - p^\alpha)/\alpha$ for $\alpha \in \{8, 10\}$, $\log(1 - p)$ with numerical clipping, and top/bottom gates at $p = 0.2$ applied to either $1 - p$ or $-\log p$. Gating uses the current probability threshold as implemented. These loss transformations change the contribution of examples at different target probabilities.

Many nonstandard objectives yield almost deterministic wrong answers: 23 of 32 non-NLL variants have mean confidence on wrong answers above 0.95. This is not true of every variant. Bottom-probability objectives can instead yield low confidence with poor accuracy. For example, Qwen2.5-3B Base with the bottom-log-probability objective has accuracy 0.3018, ECE 0.0302, Brier 0.7429, and NLL 1.3722. The low ECE is close to a low-information four-choice prediction regime. These results motivate joint reporting of calibration and prediction quality; they do not establish superiority of a new objective.

Table 21: Complete historical objective sweep on MMLU test, from the initial full report. Four untrained baselines and 36 trained variants. Top/bottom gates use threshold 0.2. All metrics are candidate-renormalized; ECE uses 10 bins.

Objective	Accuracy ↑	ECE ↓	Brier ↓	NLL ↓	Wrong conf.
Qwen3-4B Instruct					
Untrained	0.6854	0.2879	0.5937	4.4319	0.9465
NLL	0.7023	0.0550	0.3980	0.7608	0.5900
$1 - p$	0.6540	0.3439	0.6888	7.5857	0.9959
Family 8	0.6244	0.3751	0.7503	10.2595	0.9990
Family 10	0.6298	0.3698	0.7398	10.1281	0.9991
$\log(1 - p)$	0.6247	0.3747	0.7494	10.1631	0.9986
Top p	0.6186	0.3808	0.7620	10.3966	0.9991
Top log-p	0.5990	0.4005	0.8013	10.9604	0.9992
Bottom p	0.2715	0.1283	0.7368	1.3600	0.2954
Bottom log-p	0.4158	0.1454	0.7450	1.3766	0.2701
Qwen3-4B Base					
Untrained	0.6998	0.0361	0.4030	0.8036	0.5697
NLL	0.7179	0.0500	0.3823	0.7403	0.5934
$1 - p$	0.6679	0.3284	0.6584	6.9513	0.9918
Family 8	0.5328	0.4669	0.9338	12.8405	0.9995
Family 10	0.2465	0.7535	1.5069	20.8187	1.0000
$\log(1 - p)$	0.6316	0.3680	0.7362	10.1253	0.9992
Top p	0.6527	0.3467	0.6936	9.4649	0.9986

Historical objective sweep (continued)

Objective	Accuracy ↑	ECE ↓	Brier ↓	NLL ↓	Wrong conf.
Top log-p	0.6208	0.3785	0.7570	10.3067	0.9985
Bottom p	0.3390	0.0785	0.7435	1.3734	0.2590
Bottom log-p	0.2551	0.0627	0.7551	1.3956	0.3177
Qwen2.5-3B Base					
Untrained	0.5764	0.0982	0.5457	1.0154	0.5744
NLL	0.6509	0.0668	0.4530	0.8509	0.5660
$1 - p$	0.6481	0.3244	0.6658	3.5775	0.9497
Family 8	0.6364	0.3603	0.7227	9.1524	0.9945
Family 10	0.6418	0.3554	0.7126	9.1181	0.9948
$\log(1 - p)$	0.6459	0.3528	0.7061	9.5226	0.9974
Top p	0.6463	0.3461	0.6979	6.9555	0.9868
Top log-p	0.6473	0.3350	0.6808	4.5328	0.9676
Bottom p	0.3302	0.0563	0.7407	1.3677	0.2732
Bottom log-p	0.3018	0.0302	0.7429	1.3722	0.2701
Qwen2.5-3B Instruct					
Untrained	0.6372	0.3246	0.6711	3.7233	0.9304
NLL	0.6578	0.0630	0.4433	0.8349	0.5637
$1 - p$	0.6516	0.3249	0.6629	4.9676	0.9553
Family 8	0.6479	0.3489	0.6999	8.9174	0.9945
Family 10	0.6492	0.3479	0.6973	8.9468	0.9945
$\log(1 - p)$	0.6575	0.3411	0.6831	9.1912	0.9974
Top p	0.6523	0.3411	0.6854	7.9422	0.9870
Top log-p	0.6530	0.3331	0.6751	6.4631	0.9742
Bottom p	0.4214	0.1476	0.7277	1.3420	0.2704
Bottom log-p	0.4234	0.1562	0.7313	1.3495	0.2634

O.2 Changing response format and the amount of template text

The four-format Qwen2.5-3B-Instruct ladder progresses from JSON answer-only supervision to JSON full response, natural-language full response, and a longer boilerplate response. Their MMLU endpoint ECE-10 values are 0.0632, 0.0613, 0.0595, and 0.0603. On MMLU-Pro (Wang et al., 2024), endpoint ECE values are 0.0259, 0.0263, 0.0906, and 0.1461. Initial calibration varies by evaluation format, so absolute post-training numbers cannot be interpreted as matched-baseline treatment effects. The longer response is boilerplate; it is not an experiment on the calibration effects of genuine chain-of-thought reasoning.

A reported loss-dilution series adds varying amounts of template text and compares answer-only schedules of 390 and 1,170 steps. Reported MMLU endpoints include ECE 0.0614 for the shortest template, 0.0663 for answer-only at 390 steps, and 0.1129 for answer-only at 1,170 steps. The retained materials do not permit independent reconstruction of all claimed supervised-token fractions and their correlations. We therefore treat this series as historical report-level evidence only. Different training horizons also prevent a pure dilution interpretation.

O.3 Prior knowledge, answer interface, and candidate support

An eight-run, single-seed study compares Instruct/Base models, behaviorally defined Known/Unknown training subsets, and fixed/random answer interfaces for 100 updates. Known/Unknown labels come from four initial behavioral evaluations, and the subsets differ by model. They do not measure latent knowledge on a shared cross-model set. Randomizing the interface changes both the mapping and the coverage of answer tokens, so the comparison does not isolate either factor.

For Qwen on MMLU-Pro, the fixed-label Unknown condition places 97.88% of candidate probability mass on A–D although only 40.58% of gold answers fall in that subset. The random-interface condition assigns 31.42% mass to A–D. Its accuracy increases from 0.2243 to 0.3766. A descriptive support factorization assigns 0.1349 of the 0.1523 accuracy improvement and 2.0431 of the 2.1803

NLL reduction to the support component. This is a limitation of a scale-only account: global temperature cannot change the within-example ordering of candidate labels or independently restore the learned mapping to under-supported labels.

Low ECE under interface changes also requires quality checks. The Base Unknown/random MMLU-Pro result has accuracy 0.1015, Brier 0.9045, and NLL 2.329, slightly worse than the uniform ten-choice proper-score baselines. In contrast, Qwen Unknown/random on MMLU has accuracy 0.6082 and ECE 0.0440, which is not the same collapse regime. A small calibration error should be interpreted in its model, task, and answer-support context.

One nine-protocol report has an A0/A3 labeling conflict and lacks its A0 raw table, so we exclude its cross-protocol comparisons. The earlier corrective-replay proposal has no completed comparative result. It is separate from the SGCR experiments in Section 5 and supplies no evidence about longer training runs.