GALA: REWARD-GUIDED DISTILLATION ACROSS TOKENIZERS FOR ARITHMETIC REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Distilling a language model for verifiable reasoning requires reconciling two forms of supervision: a teacher's token distributions and a verifier's assessment of complete answers. When teacher and student use different tokenizers, their distributions must first be aligned; even after alignment, imitation and reward-based learning can prescribe conflicting updates. We present GALA, a training framework that combines cross-tokenizer distribution matching with group-relative verifier feedback. It constrains teacher gradients in the student's aligned logit space through conflict projection, a learned gate, and a verifier-relative norm cap, while retaining reference-solution supervision. We evaluate completed training configurations on Countdown with a Qwen3-4B teacher and Gemma-3-1B and Llama-3.2-1B students. Fixed forward achieves 44.87% and 46.12% raw accuracy in the seed-42 direction comparison; its small lead over outcome conditioning does not establish a reliable direction advantage. Supplemental component ablations report 41.00% and 41.80% strict accuracy for full GALA, exceeding unmodified KD on the same eligibility support by 5.00 and 4.90 percentage points. Fixing the gate to one while retaining projection and the norm cap yields 40.90% and 41.60%, indicating a much smaller observed contribution from learned gating. These results support gradient control over directly adding the aligned teacher gradient in the evaluated configurations, while distinguishing arithmetic correctness from successful completion.

1 INTRODUCTION

An arithmetic verifier can decide whether a completed expression reaches a target using the permitted numbers. It does not directly specify which next-token distribution a student should learn. A stronger language model provides a complementary signal: a distribution over continuations, including alternatives that the student did not sample. Combining these signals is attractive for small language models, but the combination requires more than adding a correctness score to a distillation objective.

Two difficulties arise. First, models from different families may partition the same text into different tokens and assign different meanings to vocabulary indices. Direct coordinate-wise logit matching is then undefined. Second, teacher imitation and verifier-based policy learning need not agree on a sampled trajectory. A teacher evaluates continuations under a prefix that may already contain a student's mistake; matching its conditional distribution is not identical to increasing the probability of a successful complete solution. A scalar mixture coefficient controls the overall strength of imitation, but does not identify which parts of its local gradient oppose the reward-induced update.

Consider using 2, 3, and 7 exactly once to obtain 13. A student may sample both $7 + 3 \times 2 = 13$ and $7 + 3 + 2 = 12$. The verifier distinguishes their outcomes, while teacher distributions can guide the choice of operator. This example illustrates complementary information rather than token-level error labels: the verifier does not identify the operator as the cause of failure, and a teacher is not assumed to supply a correct repair for every mistaken prefix. A useful training rule should exploit available teacher information while making its interaction with the reward-based update explicit.

We introduce GALA, a reward-guided cross-tokenizer distillation framework. It builds on an aligned hybrid distribution loss and a group-relative policy objective. The verifier supplies a sequence-level advantage, which induces a differentiable policy gradient in student logit space. We project conflicting components out of the teacher gradient, predict a local gate from detached diagnostics, and limit

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.

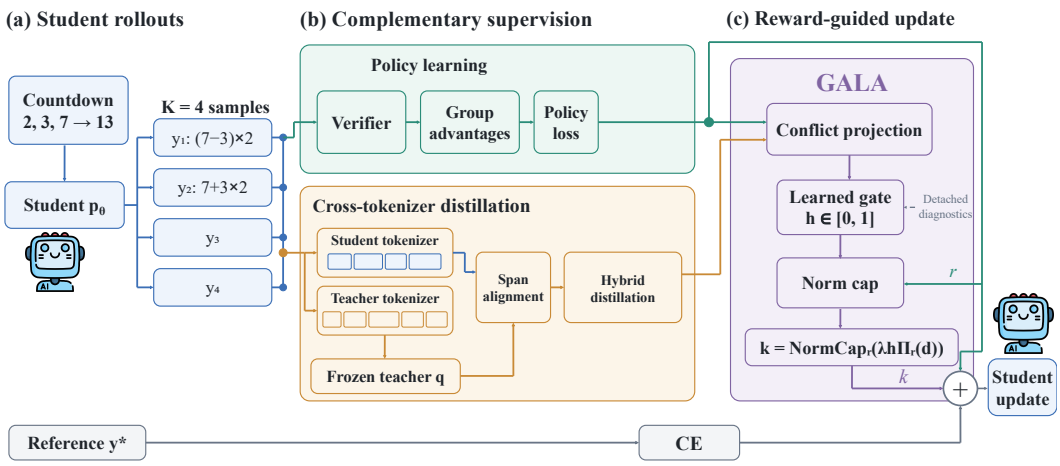


Figure 1: **The GALA training path.** Student-generated responses receive both executable outcome feedback and teacher distribution supervision. Cross-tokenizer alignment makes local distribution comparisons possible. Gate, projection, and norm control operate on student-logit gradients; reference-solution CE supplies an additional supervised signal. The verifier is not differentiated, and the teacher is frozen.

the weighted teacher-gradient norm relative to the verifier-induced gradient. The resulting vector is injected through a stop-gradient surrogate. Reference-solution cross entropy remains active, including when a rollout group provides no reward contrast. Figure 1 summarizes this training path.

We evaluate Qwen3-4B-Instruct-2507 transfer to two approximately 1B students on Countdown. Fixed forward obtains 44.87% and 46.12% raw accuracy in the single-seed direction comparison, only 0.18 and 0.32 points above outcome conditioning. Supplemental ablations compare the complete framework with unmodified aligned KD on the same support and with a fixed gate $h = 1$. They show larger observed gains from the control stack than from learning the gate, with raw and strict accuracy reported together.

The paper makes three contributions. First, it specifies a concrete mechanism for conditioning the strength and direction of cross-tokenizer teacher updates on verifier-induced policy gradients. Second, it reports a controlled direction comparison and two component ablations that test the gradient-control stack and the gate's incremental contribution, including completion-aware accuracy. Third, it analyzes the scope of the local gradient constraint and documents why mathematical correctness and valid termination must be distinguished when interpreting a reverse-direction failure. The empirical scope is Countdown with Qwen-to-Gemma and Qwen-to-Llama transfer.

2 Related Work

Distillation on student-generated trajectories. Knowledge distillation transfers a teacher's predictive distributions to a student (Hinton et al., 2015), while sequence-level distillation provides teacher-generated output targets (Kim & Rush, 2016). Training on learner-induced histories also connects to dataset aggregation in imitation learning (Ross et al., 2011). GKD brings this motivation to language-model distribution matching: the teacher scores student-generated sequences, supports alternative divergences, and can be combined with reinforcement learning (Agarwal et al., 2024). MiniLLM develops reverse-KL distillation for generative models (Gu et al., 2024a). DistiLLM studies skew-divergence objectives and adaptive reuse of student outputs (Ko et al., 2024); DistiLLM-2 couples contrastive objectives with teacher- and student-generated responses (Ko et al., 2025). Together, these studies motivate examining both the sampling distribution and the matching objective. Our direction comparison fixes the rest of the reward-guided recipe and varies the matched-vocabulary orientation within its hybrid loss.

Cross-tokenizer supervision. ULD compares distributions across different vocabularies (Boizard et al., 2024); DSKD uses dual-space alignment (Zhang et al., 2024), and approximate likelihood matching supports substantially different tokenizers (Minixhofer et al., 2026). GOLD supplies the aligned matched/unmatched hybrid construction used here (Patiño et al., 2025). More recent alternatives include a shared byte-level interface with a student byte decoder in BLD (Singh et al., 2026), contextual sequence and vocabulary mapping in CDM (Chen et al., 2025), and semantic alignment with distribution-aware transfer in SeDi (Wang et al., 2026). These methods address the comparability of teacher and student predictions. GALA instead controls a teacher gradient after alignment. Its sorted unmatched term compares probability masses without recovering token semantics; the current experiment does not establish that this alignment is preferable to those alternatives.

Verification and reward-based learning. Learned verifiers rank mathematical solutions (Cobbe et al., 2021), while process supervision evaluates intermediate steps (Uesato et al., 2022; Lightman et al., 2024). GALA uses executable outcomes with group-relative policy learning (Shao et al., 2024) and online distribution supervision. CTPD transfers preferences across tokenizers using shared spans and a DPO-style objective (Nguyen et al., 2026); GALA instead uses within-prompt standardized arithmetic rewards. VERPO combines evidence-conditioned guidance with local reward alignment and Fisher-cost control (Li et al., 2026). GALA applies Euclidean projection and a relative norm cap to a frozen external teacher's cross-tokenizer gradient, without VERPO's evidence decomposition. Findings on answer-conditioned rationale distillation also caution that correct final answers do not certify useful teacher supervision (Lee et al., 2026). Our teacher scores student-generated prefixes rather than constructing rationales toward a supplied answer.

Gradient interaction. Multi-objective formulations make task trade-offs explicit (Sener & Koltun, 2018), and GradNorm adapts their relative gradient magnitudes (Chen et al., 2018). PCGrad removes conflicting components between task gradients (Yu et al., 2020). CAGrad balances average-loss descent with local improvement across tasks (Liu et al., 2021). GALA uses an asymmetric construction: it retains the verifier-induced gradient and modifies the teacher contribution through projection, a learned gate, and a relative norm cap. This makes the roles of the two supervision sources explicit at aligned student positions. The local proposition characterizes the injected vectors, not descent of the verifier objective through shared network parameters.

Reward specification and completion. Formal analyses of reward hacking distinguish a proxy objective from the behavior it is intended to promote (Skalse et al., 2022). Our completion diagnostics make that distinction concrete for arithmetic generation: a correct expression and a well-terminated answer are separate observable properties. Reporting both helps specify the interface between the verifier, the sampled trajectory, and the distillation update. Appendix K provides corresponding audit cases.

3 Reward-guided cross-tokenizer distillation

3.1 Sequence-level verification induces a differentiable policy signal

Let x denote a problem, p_θ the student, and q a frozen teacher with a potentially different vocabulary. For each problem, we sample $K = 4$ student responses $y_1, \ldots, y_K$. The arithmetic verifier returns correctness $c_i \in \{0, 1\}$, while f_i and u_i indicate a closed answer block and valid use of the input number multiset. The reported legacy recipe uses

$$R_i = c_i + 0.1 f_i + 0.2 u_i. \tag{1}$$

Correctness requires valid number use and evaluation to the stated target. It is not granted for an equation that reaches the target using the wrong numbers. Equation 1 describes the shared V6 recipe used in the direction ablation, whose unfinished penalty is zero. Completion-aware auditing is discussed separately in Section 5 and Appendix E; its stricter rules are not assumed to underlie these historical scores.

Within each group, we compute $\bar{R} = K^{-1}\sum_i R_i$ and population standard deviation s_R. The advantage is

$$A_i = \begin{cases} (R_i - \bar{R})/(s_R + \epsilon_A), & s_R > \epsilon_A, \\ 0, & \text{otherwise}, \end{cases} \qquad \epsilon_A = 10^{-6}. \tag{2}$$

For token t in the scored student-token representation of the response, write $\ell_{it} = \log p_\theta(y_{it} \mid x, y_{i,<t})$. Following the score-function policy-gradient construction (Williams, 1992), the policy gradient is the one induced by

$$\mathcal{L}_{\mathrm{V},i} = -\frac{A_i}{T_i}\sum_{t=1}^{T_i} \ell_{it}, \tag{3}$$

with detached A_i. The implementation uses a clipped-ratio surrogate whose denominator is the detached current log probability; at evaluation of that surrogate the ratio is one and its first-order gradient agrees with Equation 3. It does not perform multiple PPO epochs against a stored old policy. Appendix A specifies this distinction.

The verifier itself is discrete and is never differentiated. Instead, for student logits z_{it}, it induces the gradient

$$r_{it} = \nabla_{z_{it}} \mathcal{L}_{\mathrm{V},i} = \frac{A_i}{T_i}\big(p_\theta(\cdot \mid x, y_{i,<t}) - e_{y_{it}}\big), \tag{4}$$

where $e_{y_{it}}$ is the one-hot sampled token. A single sequence advantage acts on all tokens; this is not process supervision or a verifier-derived localization of an error. Furthermore, an incorrect response can have positive advantage if its shaped reward exceeds that of the other candidates. The method uses relative group feedback rather than treating every incorrect candidate as a negative-advantage example.

3.2 A HYBRID TEACHER LOSS ACROSS DIFFERENT TOKENIZERS

We score the same sampled response under the teacher and student tokenizers. Shared text boundaries select a verifier-relevant window, capped at 64 tokens per side; a closed answer block is preferred, with a suffix fallback when no block exists. Within this window, greedy matching of normalized decoded token pieces forms groups covering corresponding text. Multiple token distributions in one group are merged by a normalized product. This is the aligned hybrid construction inherited from the GOLD path (Patiño et al., 2025), rather than a learned semantic alignment or a new vocabulary transport model.

Let p_a and q_a be the merged distributions at aligned group a, and let $\mathcal{M}$ contain matched teacher–student vocabulary pairs. For the forward configuration, the matched contribution is

$$D_a^{\rightarrow} = \sum_{(v_T, v_S)\in\mathcal{M}} q_a(v_T) \log \frac{q_a(v_T)}{p_a(v_S)}. \tag{5}$$

Reverse swaps the roles in each summand: $D_a^{\leftarrow} = \sum_{(v_T,v_S)\in\mathcal{M}} p_a(v_S)\log[p_a(v_S)/q_a(v_T)]$. Let p_a^{U} and q_a^{U} contain the unmatched probabilities. After sorting each in descending order and zero-padding to equal width, the hybrid loss is

$$\mathcal{L}_{\mathrm{D},i} = \frac{1}{|\mathcal{A}_i|}\sum_{a\in\mathcal{A}_i}\left[D_a^{b_i} + \left\|\mathrm{sort}(p_a^{\mathrm{U}}) - \mathrm{sort}(q_a^{\mathrm{U}})\right\|_1\right]. \tag{6}$$

Both contributions have coefficient one. The matched subsets are *not* independently renormalized; Equation 5 is a restricted KL contribution, not a standalone divergence between normalized distributions. In particular, it need not be nonnegative. Throughout the paper, "forward" and "reverse" identify this matched-term orientation inside Equation 6, not pure full-vocabulary KL objectives.

Fixed forward uses $b_i = \rightarrow$ for every supported candidate; fixed reverse uses $b_i = \leftarrow$. Outcome conditioning uses reverse for $c_i = 1$ and forward otherwise. All retain the correctness-dependent base coefficient

$$\lambda_i = 0.2\,[0.25c_i + (1 - c_i)]. \tag{7}$$

This inherited heuristic assigns four times the base KD weight to incorrect responses, emphasizing teacher supervision where the sampled answer fails. It is not an optimized ratio: gating and capping determine the eventual magnitude. Fixing direction retains this correctness dependence.

3.3 CONTROLLING THE TEACHER GRADIENT WITH VERIFIER FEEDBACK

At each student position in the aligned window, let $d_{it} = \nabla_{z_{it}} \mathcal{L}_{\mathrm{D},i}$. We compare it with r_{it} from Equation 4. For clarity, the ideal conflict-removal operation is

$$\Pi_r(d) = \begin{cases} d - \dfrac{\langle d, r \rangle}{\|r\|_2^2} r, & \langle d, r \rangle < 0, \\ d, & \text{otherwise.} \end{cases} \tag{8}$$

This operation is asymmetric: it modifies the distillation direction while leaving the verifier-induced direction intact. The implementation uses FP32 arithmetic, a denominator floor, and a small positive margin on conflicting positions; these numerical details and the exact scope of the ideal property appear in Appendix B.

A two-layer network predicts $h_{it} \in [0, 1]$ from detached local diagnostics. Its features include the hybrid loss, correctness, group advantage, sampled-token surprisal, response position, and alignment-group width. For eligible candidates it is trained to predict whether the original teacher and verifier gradients agree, using binary cross entropy and a small adjacent-token smoothness penalty. The pre-projection target preserves information that an ideal post-projection sign test would erase. The gate attenuates the remaining vector; its prediction target is agreement, not correctness or future reward. Detached features and gates prevent its supervision loss from updating the student.

Let e_i indicate candidate eligibility, including nonzero group advantage. In the legacy outcome-enabled direction suite, incorrect candidates can also be eligible; this includes fixed reverse under a direction override. The norm-controlled, weighted teacher update is

$$v_{it} = e_i h_{it} \Pi_{r_{it}}(d_{it}), \tag{9}$$

$$s_{it} = \min\left(1, \frac{\rho \|r_{it}\|_2}{\lambda_i \|v_{it}\|_2}\right), \qquad k_{it} = \lambda_i s_{it} v_{it}, \qquad \rho = 1, \tag{10}$$

with $k_{it} = 0$ when the required gradient or denominator is zero. Candidate eligibility and the cap together prevent KD from becoming the only local update on a zero-advantage trajectory. A learned gate controls attenuation, whereas the projection and cap impose explicit local constraints; these roles should not be conflated.

Proposition 1 (Ideal local alignment and magnitude)**.** *With the exact projection in Equation 8, nonnegative gates and coefficients, and Equation 10, each position satisfies* $\langle r_{it}, k_{it} \rangle \geq 0$ *and* $\|k_{it}\|_2 \leq \rho \|r_{it}\|_2$. *Consequently,* $\langle r_{it}, r_{it} + k_{it} \rangle \geq \|r_{it}\|_2^2$.

This is a geometric characterization of the ideal injected vectors. For stacked logits with Jacobian $J = \partial z / \partial \theta$, parameter-gradient agreement instead depends on $r^T J J^T k$, whose sign need not match $r^T k$. It therefore gives neither parameter-space nonconflict nor monotonic reward improvement. Appendix B supplies the proof, a counterexample, and numerical qualifications.

3.4 STUDENT UPDATE AND REFERENCE FALLBACK

The teacher vector is injected through the surrogate

$$\mathcal{L}_{\mathrm{inject},i} = \sum_{t \in \mathcal{W}_i} \langle z_{it}, \mathrm{sg}(k_{it}) \rangle, \tag{11}$$

whose logit gradient is exactly the detached vector k_{it}. The student objective is

$$\mathcal{L}_{\mathrm{student}} = \frac{1}{K} \sum_{i=1}^{K} (\mathcal{L}_{\mathrm{V},i} + \mathcal{L}_{\mathrm{inject},i}) + 0.1 \alpha_x \mathcal{L}_{\mathrm{CE}}(x, y^*), \qquad \alpha_x = 0.25 + 0.75 \mathbb{I}\Big[\max_i c_i = 0\Big]. \tag{12}$$

CE therefore remains present even when at least one candidate is correct, and receives its full anchor scale for all-wrong groups. If every candidate has identical shaped reward, all A_i are zero: policy and admitted KD contributions vanish, but reference learning remains. Uniform correctness alone is not sufficient for this condition, since format and number-use rewards can differ within an all-wrong group.

The teacher is frozen; the gate is optimized separately. Rollout prefetching stays inside the student-update boundary, so all prefetched candidates within a cycle use the same student weights. Algorithm 1 gives the full update sequence in the appendix.

4 EXPERIMENTAL SETTING

Task, models, and data. Countdown asks a model to construct an arithmetic expression that uses the supplied number multiset exactly once and evaluates to a target. Evaluation checks the expression and its stated target, rather than exact string agreement with a reference. We use the `HuggingFaceTB/Countdown-Task-GOLD` data revision listed in Appendix C: 30,441 training examples and 10,000 test examples. The offline training references come from the dataset's verified Qwen2.5-7B-Instruct configuration; the online teacher is Qwen3-4B-Instruct-2507 (Yang et al., 2025). Students are Gemma-3-1B-it (Team et al., 2025) and Llama-3.2-1B-Instruct (Grattafiori et al., 2024). Thus the online teacher and the model that produced the reference corpus are different.

Training and hardware. The completed experiments were reported as run on four NVIDIA H200 GPUs. The V6 main recipe uses seed 42, four candidates per prompt, sampling temperature 0.7, top-p 0.95, a 512-token prompt limit, and a 1,024-new-token training limit. The nominal four-rank schedule has microbatch size 1, accumulation 2, and one student optimizer update per eight data steps: eight prompts per data step and 64 per complete update cycle. Training ends at data step 4,757. Student computation uses BF16, with CPU FP32 master parameters and AdamW state (Loshchilov & Hutter, 2017), learning rate 2×10^{-6}, cosine scheduling, and warmup ratio 0.02. The Gemma schedule retains a 600-step initial stage before continuation; Llama trains directly to the final step. Appendix C records the remaining recipe details. We report neither measured H200 throughput nor a compute-matched efficiency advantage.

Baselines and controlled direction ablation. We compare three GALA variants: fixed forward, fixed reverse, and outcome-conditioned direction selection. Within each student family, they use the same reward, EOS/termination criteria, data, training schedule, and remaining method settings; only the KL direction rule changes. All three retain verifier feedback, alignment, gate, gradient projection, norm control, and reference CE. Their direction labels specify the orientation of the matched-vocabulary term, not pure KL distillation. This provides a controlled direction ablation under the shared V6 recipe. SFT is a separate baseline trained on reference solutions without an online teacher or policy objective. Later completion-aware repairs are not retrospectively assigned to these scores.

Evaluation. The original direction-suite final scores use checkpoint 4,757 and 10,000 Countdown test problems, with vLLM (Kwon et al., 2023) greedy decoding, seed 42, a 512-token input limit, and a 2,048-new-token generation limit. Accuracy refers to the reported mathematical verifier, not to an additional requirement that the response terminate cleanly. The comparison uses the same final step for both students, including Llama, whose outcome score at step 4,000 is slightly higher. Intermediate outcome checkpoints describe the trajectory. The direction ranking was identified after inspecting this same test set; there was no held-out validation split for direction selection. We therefore report all three policies at the common final checkpoint as an exploratory comparison. Each original direction configuration has one seed-42 training run. Appendix N.1 reports conditional test-sampling intervals; training-seed variance and paired differences require additional records.

5 RESULTS

5.1 SINGLE-SEED FINAL MATHEMATICAL ACCURACY

Table 1 gives the final raw accuracies for both students. Relative to the reported SFT baselines, its gains are 14.79 percentage points for Gemma and 12.63 for Llama. The outcome-conditioned configuration also exceeds those baselines, by 14.61 and 12.31 points.

The forward–outcome differences are small: 0.18 point for Gemma and 0.32 for Llama, corresponding to 18 and 32 net correct answers out of 10,000. Fixed forward has the larger point estimate in each completed run. These small gaps do not establish superiority or equivalence between the policies; the direction ranking remains descriptive.

Fixed reverse performs differently across the students. It is below SFT for Gemma by 5.84 points, but above SFT for Llama by 10.22 points. This asymmetry is incompatible with a blanket conclusion

Table 1: **Final raw mathematical accuracy (%).** Local results use 10,000 problems, seed 42, and checkpoint 4,757. GOLD is an external reported reference, not a rerun. The GALA variants share settings except for matched-term direction. Bold marks the largest observed value, not statistical significance.

Configuration	Gemma-3-1B-it	Llama-3.2-1B-Instruct
SFT	30.08	33.49
GOLD (external report)	3.05	42.35
GALA fixed forward	**44.87**	**46.12**
GALA outcome-conditioned	44.69	45.80
GALA fixed reverse	24.24	43.71

Table 2: **Supplemental component ablations (accuracy, %).** Raw and strict scores are reported together. Full GALA is the reference configuration. Differences are observed point estimates, not significance tests.

	Gemma		Llama	
Configuration	Raw	Strict	Raw	Strict
Same-support unmodified KD	41.30	36.00	42.70	36.90
Fixed gate ($h = 1$)	44.70	40.90	46.00	41.60
Full GALA	**44.87**	**41.00**	**46.12**	**41.80**

that reverse-direction training cannot benefit approximately 1B models. It also motivates examining generation behavior before assigning the Gemma gap solely to the mathematical form of a divergence.

Comparison with GOLD. The official GOLD report gives 3.05% for Gemma and 42.35% for Llama (Patiño et al., 2025). Fixed-forward scores are higher by 41.82 and 3.77 percentage points, and outcome-conditioned scores by 41.64 and 3.45 points. The experiment owner confirmed correspondence of the stated settings, but GOLD was not rerun in this codebase. These are differences from an external reference; the large Gemma gap cannot be assigned to a specific mechanism without a direct reproduction. Appendix L separates score provenance from protocol confirmation.

5.2 Gradient-control and learned-gate ablations

Table 2 reports two supplemental interventions. *Same-support unmodified KD* retains the verifier, aligned loss, correctness-dependent KD coefficient, eligibility mask, and reference CE, but injects $k_{it} = \lambda_i e_i d_{it}$ without projection, gate, or cap. *Fixed gate* sets $h_{it} = 1$ while retaining projection and the cap. Same support denotes the same eligibility rule, not identical sampled trajectories across training runs.

Full GALA exceeds unmodified KD by 3.57/3.42 points in raw accuracy and 5.00/4.90 points in strict accuracy for Gemma/Llama. With $h = 1$, the corresponding gains remain 3.40/3.30 raw and 4.90/4.70 strict points. Thus most of the observed improvement is retained by projection and norm control without a learned gate. Learning the gate adds 0.17/0.12 raw and 0.10/0.20 strict points; these small increments do not establish a statistically reliable gate advantage. The comparison tests the control stack jointly, without separating projection from the cap. Appendix N.2 distinguishes completed controls from remaining interventions.

5.3 The outcome trajectories have different late-training behavior

Figure 2 shows outcome-conditioned trajectories only. Gemma rises from 11.31% at step 600 to 44.69%, with 6.42 points gained after step 3,000. Llama rises from 30.55% at step 1,000 to 45.80%, gaining only 0.57 point after step 3,000. Its step-4,000 score is 45.83%; we retain the final checkpoint rather than selecting that maximum. The different late-training profiles do not establish a direction-dependent convergence advantage. Appendix D lists all values.

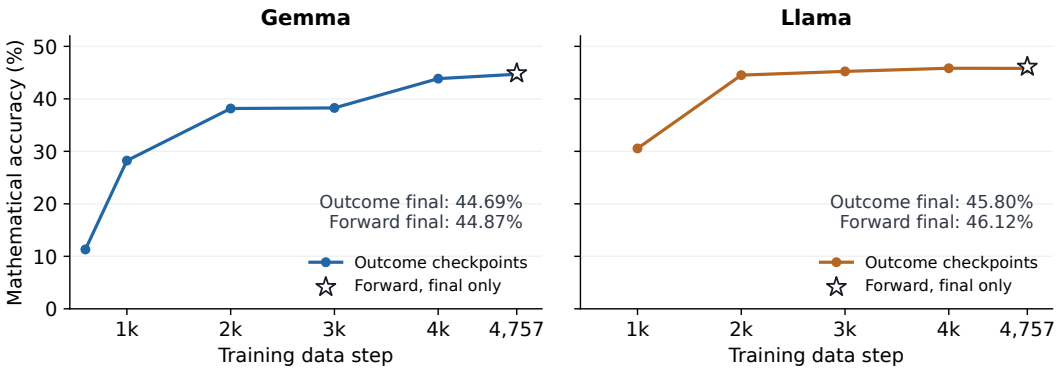


Figure 2: **Completed outcome-conditioned checkpoint evaluations.** Lines connect only supplied measurements; no Llama checkpoint at step 600 was saved. Stars mark the fixed-forward final scores and are not forward training curves. All points use 10,000 test problems. These curves describe the outcome runs and do not establish a convergence-speed advantage over fixed forward.

Table 3: **Late-training diagnostics from the fixed-direction suite.** Candidate correctness and the fraction without EOS are averages over data steps 4,258–4,757, using sampled training rollouts capped at 1,024 new tokens. They are not greedy test accuracies.

Student	Direction	Candidate correct (%)	Without EOS (%)
Gemma	Forward	70.90	7.80
Gemma	Reverse	35.81	99.71
Llama	Forward	71.79	15.40
Llama	Reverse	69.71	4.09

5.4 Mathematical accuracy and completion are distinct

The supplied diagnostics reveal a substantial termination failure in Gemma fixed reverse (Table 3). In the last 500 training steps, 99.71% of sampled candidates do not emit EOS. In its final greedy test, 99.82% of outputs reach the 2,048-token generation limit. The recorded failure case includes an arithmetically correct answer followed by approximately 1,626 newline characters. Under the legacy reward, correctness does not require EOS, and the unfinished penalty is zero; such an output can therefore receive mathematical correctness credit despite invalid completion behavior.

This failure was not preceded by uniformly incorrect rollouts. The report records 70.75% candidate correctness at steps 2,601–2,700 and 65.19% at steps 2,901–3,000; the latter interval already has a 71.09% no-EOS rate. Thus correctness and completion behavior can diverge before the final accuracy deteriorates. The 1,626-newline diagnostic does not imply that a training rollout exceeded its 1,024-token cap: training and final evaluation have different generation limits.

Completion also matters for fixed forward: sampled no-EOS rates are 7.80% for Gemma and 15.40% for Llama. These training rates cannot adjust final accuracy. Table 2 now reports strict scores for full GALA and both controls; separate final natural-end and terminal-format rates, and strict scores for the other direction policies, remain unavailable.

The same explanation cannot simply be applied to Llama. Its fixed-reverse no-EOS rate is only 4.09%, lower than the 15.40% of fixed forward, while fixed forward still has higher final mathematical accuracy. Termination is consequently an important confound in the Gemma comparison, not a universal explanation of the direction ranking. Likewise, allowing wrong candidates to receive reverse KD in a direction-override configuration is a plausible contributor to degradation, but these logs do not establish causality. Outcome-conditioned V6 already assigns wrong candidates to forward KD; the override failure must not be attributed indiscriminately to that configuration.

6 DISCUSSION

What the component comparison establishes. The same-support control retains online KD, verifier feedback, and CE, making it more specific to gradient modification than SFT or external GOLD comparisons. Full GALA has higher raw and strict scores than this control for both students. Projection and the cap with $h = 1$ retain most of the observed gain; the learned gate adds only 0.10/0.20 strict points. This supports the combined constraints in the tested setting without establishing a reliable gate advantage or isolating projection from norm control.

Direction and completion. The small forward–outcome differences remain a descriptive, test-inspected single-seed ranking. Strict evaluation now covers the component comparison, but does not establish a completion-aware ranking of all three directions. The study remains limited to Countdown, one teacher, two approximately 1B students, and group size four. Replicated training and per-item paired records are needed to quantify uncertainty in both comparisons.

7 CONCLUSION

GALA combines cross-tokenizer teacher distributions, local gradient control, verifier feedback, and reference supervision. In the supplemental ablations, full GALA reaches 41.00%/41.80% strict accuracy for Gemma/Llama, compared with 36.00%/36.90% for same-support unmodified KD and 40.90%/41.60% with $h = 1$. The results support the control stack over directly adding KD in these configurations; the learned gate's incremental benefit is small. The direction study and completion diagnostics delimit what the current evidence establishes.

REPRODUCIBILITY STATEMENT

The appendix specifies the loss, gate inputs and targets, local gradient constraints, data revision, nominal V6 training schedule, and evaluation protocol. Appendices G–K provide a worked update, additional derivations, the ablation design, restoration requirements, and completion-audit cases. Tables and figures use the accompanying aggregate results or explicitly labeled constructed examples. We distinguish completed result series from later implementation repairs and do not substitute results from an unfinished run. The three direction variants share the same reward, termination criteria, and training configuration, with only the KL direction rule varied (Section 4). GOLD scores are external reported results with author-confirmed configuration correspondence, rather than a reproduction in our codebase (Appendix L). Appendix N gives aggregate-data uncertainty calculations and separates available evidence from protocols requiring new records or runs.

AI USE STATEMENT

AI Assistance: Yes, to aid or polish writing. Details are described in the paper.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for llms. *arXiv preprint arXiv:2402.12030*, 2024.

Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. Enhancing cross-tokenizer knowledge distillation with contextual dynamical mapping. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8005–8018, 2025.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024a.

Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Miniplm: Knowledge distillation for pre-training language models. *arXiv preprint arXiv:2410.17215*, 2024b.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. Distillm-2: A contrastive approach boosts the distillation of llms. *arXiv preprint arXiv:2503.07067*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Jungseob Lee, Seungyoon Lee, Suhyune Son, Dongyub Jude Lee, Sungbin Han, Sugyeong Eo, and Heuiseok Lim. Answer-conditioned chains of thought degrade verifiable-reasoning distillation in large language models. *arXiv preprint arXiv:2607.14552*, 2026.

Haijiang Li, Chengyu Lv, Yi Zhang, Zhibing Zhang, Rui Qian, Yuchen Zhang, Xiaofan Zhang, Mingshan Wang, Xiaofei Jing, Yu Tong, et al. Verpo: Verified evidence regularized policy optimization. *arXiv preprint arXiv:2609.06100*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. *Advances in Neural Information Processing Systems*, 38: 79297–79326, 2026.

Truong Nguyen, Phi Van Dat, Ngan Nguyen, Linh Ngo Van, Trung Le, and Thanh Hong Nguyen. Ctpd: cross tokenizer preference distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37783–37790, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family. https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation, 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Avyav Kumar Singh, Yen-Chen Wu, Alexandru Cioba, Alberto Bernacchia, and Davide Buffelli. Cross-tokenizer llm distillation through a byte-level interface. In *Proceedings of the Second Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 84–96, 2026.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in neural information processing systems*, 35:9460–9471, 2022.

Sharath Turuvekere Sreenivas, Adithyakrishna Venkatesh Hanasoge, Mingyu Yang, Ali Taghibakhshi, Saurav Muralidharan, Ashwath Aithal, and Pavlo Molchanov. X-token: Projection-guided cross-tokenizer knowledge distillation. *arXiv preprint arXiv:2605.21699*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Huazheng Wang, Yongcheng Jing, Haifeng Sun, Jingyu Wang, Jianxin Liao, Leszek Rutkowski, and Dacheng Tao. Bridging the tokenizer gap: Semantics and distribution-aware knowledge transfer for unbiased cross-tokenizer distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33494–33502, 2026.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

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.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18164–18181, 2024.

A TRAINING ALGORITHM AND IMPLEMENTATION DETAILS

Algorithm 1 GALA training for one prompt group; accumulate student gradients to the configured update boundary.

Require: Frozen teacher q, student p_θ, gate h_ϕ, reference (x, y^*), direction rule b

Sample $K = 4$ responses from the student with the configured generation policy.
Verify each response; compute c_i, f_i, u_i, R_i and group advantages A_i.
for each candidate y_i **do**
    Score student token probabilities; form the verifier policy loss $\mathcal{L}_{\mathrm{V},i}$.
    if candidate has nonzero advantage and is eligible for KD **then**
        Score the same response with the teacher; align a common text window.
        Construct the hybrid matched/unmatched loss with direction b_i.
        Compute teacher gradient d_{it} and verifier-induced gradient r_{it}.
        Project conflicting teacher components and predict detached gates.
        Cap the weighted teacher norm to obtain k_{it}.
        Add the surrogate $\sum_t \langle z_{it}, \mathrm{sg}(k_{it}) \rangle$ to the student loss.
        Accumulate gate supervision from pre-projection gradient agreement.
    else
        Set the KD contribution to zero; retain the policy loss.
    end if
end for
Update the gate using its detached-feature classification loss.
Backpropagate the averaged policy and KD-surrogate objective.
Backpropagate reference CE with the group-dependent anchor coefficient.
At an update boundary, clip student gradients and update FP32 master parameters.
Cast updated master parameters into the BF16 student; clear accumulated gradients.

Policy surrogate. The executable implementation forms $a_{it} = \exp(\ell_{it} - \mathrm{sg}(\ell_{it}))$ and minimizes

$$-\frac{1}{T_i} \sum_t \min\{a_{it} A_i, \mathrm{clip}(a_{it}, 1-\varepsilon, 1+\varepsilon) A_i\}, \qquad \varepsilon = 0.2. \tag{13}$$

The ratio has numerical value one in the same forward pass. Its derivative at that point produces Equation 3; the scalar loss value is not the negative log-likelihood expression itself. This implementation should not be described as multi-epoch PPO (Schulman et al., 2017) or as correcting sampling-policy mismatch with stored behavior-policy likelihoods. Rollouts use temperature/top-p sampling, while the student scoring distribution is the model distribution used by the training loss.

Alignment and distribution merging. The implementation first selects teacher/student windows covering shared text boundaries near the final answer. Greedy matching then accumulates normalized decoded token pieces until corresponding groups match. A group G containing more than one token is represented by

$$\bar{p}_G(v) = \frac{\prod_{t \in G} \max(p_t(v), 10^{-8})}{\sum_w \prod_{t \in G} \max(p_t(w), 10^{-8})}. \tag{14}$$

Singleton groups retain their token distribution. This operation provides a comparable aligned distribution for the inherited hybrid loss; it is not the probability of a text span under an exact cross-tokenizer sequence model. Matched vocabulary entries are indexed through the supplied mapping, and unmatched probabilities retain their original mass before sorting and padding. We do not reinterpret the sorted unmatched comparison as a semantic alignment of unmatched token identities.

Gate. The gate is a six-input, 32-hidden-unit MLP with Tanh and sigmoid output. Its last layer is initialized to zero, yielding an initial output of 0.5. Its inputs are: $\log(1 + \max(L_a, 0))$ clipped at 10; verifier correctness; group advantage; sampled-token surprisal clipped at 20 and divided by 20; relative response position; and aligned group width clipped at 4 and divided by 4. Aligned-group quantities are broadcast to their member student positions. All features are detached. For eligible

nonzero-advantage candidates, the target is $\mathbb{I}[\langle d, r\rangle \geq 0]$. Binary cross entropy is supplemented by 0.01 times the mean squared difference of neighboring gates. The gate learning rate is 10^{-3}; gate gradients are clipped to norm 1. This construction supplies a trainable attenuator within the complete framework evaluated in the main results.

Precision and updating. Teacher scoring is performed without parameter gradients. Student forward computation uses BF16, while important probability reductions and gradient comparisons use FP32. The optimizer maintains student master parameters and AdamW state (Loshchilov & Hutter, 2017) in CPU FP32; updated masters are copied back to the BF16 model. Student gradient clipping uses norm 1. This is an implementation choice that preserves small accumulated updates, not a measured efficiency claim. Saving and resuming require student weights, master parameters, optimizer and scheduler states, gate and gate-optimizer state, the data-stream position, and per-rank random-number-generator states.

B Scope and proof of the local gradient property

Proof of Proposition 1. Suppress position indices. If $\langle d, r\rangle \geq 0$, then $\Pi_r(d) = d$ and its inner product with r is nonnegative. Otherwise, for $r \neq 0$,

$$\langle r, \Pi_r(d)\rangle = \langle r, d\rangle - \frac{\langle d, r\rangle}{\|r\|^2}\|r\|^2 = 0.$$

Multiplication by the nonnegative eligibility, gate, coefficient, and cap preserves the sign. For a nonzero gated vector v, Equation 10 gives $\|k\| = \min(\lambda\|v\|, \rho\|r\|) \leq \rho\|r\|$. The zero-vector case holds by definition. Finally, $\langle r, r + k\rangle = \|r\|^2 + \langle r, k\rangle \geq \|r\|^2$. □

Numerical implementation. For a conflicting pair, the implementation subtracts

$$\frac{\langle d, r\rangle - \eta\|d\|\|r\|}{\max(\|r\|^2, \epsilon_P)}\, r, \qquad \eta = 10^{-2}, \quad \epsilon_P = 10^{-12}. \tag{15}$$

When the denominator floor is inactive and arithmetic is exact, the post-projection dot product is $\eta\|d\|\|r\|$, providing a small positive margin. If the floor is active, this identity does not hold; finite-precision reductions can also produce small negative values. The implementation retains the projected vector in FP32 and suppresses KD where the verifier-gradient norm is below the trust rule's threshold. The ideal proposition is consequently not asserted as an exact invariant for every floating-point value produced by the code.

Why this is not a parameter-space or reward-improvement guarantee. Let $J = \partial z/\partial\theta$. Logit-space agreement $r^T k \geq 0$ does not imply parameter-gradient agreement $r^T J J^T k \geq 0$. For example, take

$$r = \begin{bmatrix}1\\0\end{bmatrix}, \qquad k = \begin{bmatrix}0\\1\end{bmatrix}, \qquad J = \begin{bmatrix}1 & 0\\-2 & 1\end{bmatrix}.$$

Then $r^T k = 0$ and $\|k\| = \|r\|$, yet $(J^T r)^T (J^T k) = -2$. Even the combined parameter-space direction has $(J^T r)^T J^T (r + k) = -1$ in this constructed example. This is an algebraic counterexample, not an experimental observation. It explains why shared parameters, optimizer transformations, the CE anchor, and changes in the generated trajectories preclude deriving monotonic task improvement from the local projection property alone.

C Data and training protocol

The fixed data revision is `a858e1a9b61f8687b5ecd513566499dcdfea5c39` of `HuggingFaceTB/Countdown-Task-GOLD`. Training uses configuration `verified_Qwen2.5-7B-Instruct` and evaluation uses `test`. The processed training and test counts are 30,441 and 10,000. Their locally recorded loaded fingerprints are `66033884fb7ba486` and `8f09b3de28bd004c`, respectively. The dataset is available at https://huggingface.co/datasets/HuggingFaceTB/Countdown-Task-GOLD.

Table 4: Shared V6 training recipe for the three GALA direction variants. Within each student family, reward, termination criteria, and training settings are held fixed while the KL direction rule varies.

Item	Setting
Online teacher	`Qwen/Qwen3-4B-Instruct-2507`
Students	`google/gemma-3-1b-it` `meta-llama/Llama-3.2-1B-Instruct`
Reported hardware	Four NVIDIA H200 GPUs
Seed / final data step	42 / 4,757
Student ranks / microbatch / accumulation	4 / 1 / 2
Student update interval / rollout prefetch	8 data steps / 8 data steps
Candidates per prompt	4
Sampling temperature / top-p	0.7 / 0.95
Prompt / rollout token limits	512 / 1,024 new tokens
Student optimizer	AdamW with CPU FP32 masters
Student learning rate / weight decay	2×10^{-6} / 0.1
Schedule / warmup ratio	Cosine / 0.02
Gate learning rate / hidden dimension	10^{-3} / 32
Base KD coefficient / norm ratio	0.2 / 1.0
Correct / incorrect KD scale	0.25 / 1.0
Alignment window	At most 64 tokens per side
Reference CE weight / anchor floor	0.1 / 0.25
Policy clip parameter	0.2

The nominal schedule contains 38,056 prompt occurrences and 595 student optimizer updates before any length-based rejection. There are 594 full 64-prompt cycles and a final 40-prompt cycle; the corresponding nominal number of sampled candidates is 152,224. These are arithmetic consequences of the stated schedule, not measured accepted-example or token counts. In the shared Gemma recipe, the initial stage ends at step 600 with a scheduler horizon of 75 updates; continuation uses horizon 595 while restoring optimizer, data, and RNG state. Llama uses horizon 595 directly. The Gemma two-stage schedule is retained across its direction variants; it is not equivalent by construction to using a single uninterrupted cosine schedule.

For final evaluation, the reported protocol is one greedy response per item, seed 42, maximum prompt length 512, and maximum generated length 2,048, using vLLM and the student chat template. SFT uses reference-answer CE without online KD or group-relative rollouts. This comparison measures the complete training objectives. The three GALA variants instead share the same reward, termination criteria, and training configuration, allowing a controlled comparison of direction policies. The reported scores correspond to the run count and seed specified in Section 4.

D Complete reported outcome checkpoint series

Table 5: Mathematical accuracy (%) on the full 10,000-problem test set for the outcome-conditioned run series. "Not saved" is missing data, not a zero score. The final checkpoint is used in the main results.

Data step	Gemma	Llama
600	11.31	Not saved
1,000	28.23	30.55
2,000	38.16	44.52
3,000	38.27	45.23
4,000	43.85	45.83
4,757	44.69	45.80

Counts and comparison design. With 10,000 examples, the final fixed-forward counts are 4,487 and 4,612, while the outcome counts are 4,469 and 4,580. The corresponding net differences are 18 and 32 correct answers. Results are point accuracies from one seed-42 training run per configuration. The controlled direction ablation varies the matched-term direction; gate, projection, norm control, and reference CE are retained together. The SFT and GOLD comparisons assess the complete training recipes. This design specifies the unit of evidence for interpreting the reported gains.

E MATHEMATICAL CORRECTNESS AND COMPLETION AUDITING

The legacy mathematical verifier extracts an answer block and checks number usage and target evaluation. This can distinguish a valid expression from an invalid one, but does not by itself require a single terminal answer block or a naturally ended response. The legacy unfinished penalty is further conditioned on absence of a closed answer, so simply changing its coefficient would not necessarily penalize a correct closed answer followed by an unfinished tail.

A separate completion-aware audit can define

$$c_{\text{strict}} = c_{\text{raw}} \wedge e_{\text{natural}} \wedge f_{\text{terminal}},$$

where e_{natural} is derived from actual model-specific EOS/EOT tokens or trusted generation termination metadata, and f_{terminal} requires exactly one nonempty complete answer block followed by at most eight whitespace characters. Whitespace must be inspected before stripping. An immediate EOS with no answer is not success; an EOS generated exactly at the output cap can be a natural end; reaching the cap without an ending token is not. These distinctions address separate failure modes and should not be collapsed into a test for the string `</answer>`.

Later completion-aware code also permits a negative reward for unfinished or malformed outputs, zero KD on invalid completions, and reverse-KD admission only for strictly correct candidates. These are changes to the training recipe and to fixed-reverse candidate support, not merely new logging fields. This manuscript does not attach the existing 44.87/46.12 or 44.69/45.80 scores to that repaired recipe, reports the supplemental strict scores separately in Table 2, and does not claim that the repair has established the cause of degradation. The reported termination diagnostics are evidence of a mismatch between mathematical success and generation behavior in a historical run.

F RESULT PROVENANCE AND EVIDENCE SUMMARY

The original results comprise the SFT/fixed-direction suite and the outcome-conditioned checkpoint trajectories. The authors subsequently supplied two component-ablation configurations and strict scores for the full GALA reference (Table 2); these aggregates are recorded separately in `data/component_ablations.json`. Hardware is reported as four H200 GPUs. The machine-readable tables reproduce those aggregates exactly. Configuration parity across the three direction variants was confirmed by the experiment owner: rewards, termination criteria, and other training settings are shared, with only the direction rule varied. The local V6 code was inspected to specify the algorithm and distinguish legacy from completion-aware implementations; local reference hashes are not presented as independently verified hashes of all H200 executions. No measurements from unfinished local reruns are mixed into the results. The GOLD baseline is attributed to its official report in Appendix L; the experiment owner also confirmed its training and evaluation configuration parity with the main comparison.

G A WORKED FOUR-CANDIDATE UPDATE

This section illustrates the reported reward and update equations with constructed examples. The responses and probabilities below are explanatory examples, not sampled model outputs or additional experimental measurements.

G.1 FROM MATHEMATICAL OUTCOMES TO RELATIVE ADVANTAGES

Consider the prompt "Use 2, 3, and 7 exactly once to obtain 13." Table 7 gives a possible group of four responses. The first expression is correct. The second respects number usage but misses

Table 6: Evidence supplied by the experiments and analysis.

Finding	Supporting evidence
Fixed forward attains the highest final accuracy in both students.	The three-direction comparison under shared training and evaluation settings.
GALA scores exceed SFT and the external GOLD reference.	Final aggregate accuracies; GOLD was not reproduced locally.
Forward–outcome differences are small in the completed runs.	Net differences of 18 and 32 correct answers; one training seed.
Completion behavior adds information to arithmetic accuracy.	Sampled-rollout EOS rates and the Gemma fixed-reverse generation diagnostics.
The ideal projected logit update satisfies alignment and norm bounds.	Proposition 1, its proof, and the explicit implementation conditions.

the target. The third states the target without using the required numbers. The fourth contains no parseable answer. We assume the first three expressions are enclosed in complete answer tags and the fourth receives no format or rules credit. The mean reward is 0.425, and the population variance is

Table 7: Constructed example under the shared V6 reward. c, f, and u denote mathematical correctness, closed-answer format, and number-use validity. Advantages are rounded to three decimals.

Candidate	Answer content	c	f	u	R	A
1	$7 + 3 \times 2 = 13$	1	1	1	1.3	1.694
2	$7 + 3 + 2 = 12$	0	1	1	0.3	-0.242
3	13	0	1	0	0.1	-0.629
4	No answer	0	0	0	0.0	-0.823

$$s_R^2 = \tfrac{1}{4}\left(0.875^2 + (-0.125)^2 + (-0.325)^2 + (-0.425)^2\right) = 0.266875.$$

Equation 2 gives the displayed advantages. For nonempty scored responses, their signs determine whether the policy term increases or decreases sampled-token likelihoods at the current scoring point. An empty response supplies no response-token policy gradient. This is a sequence-wide signal: the operator in candidate 2 does not receive an independently verified error label.

Under fixed forward, all candidates that pass the shared eligibility conditions request the forward matched term. Under outcome conditioning, candidate 1 requests reverse and the others request forward. The base KD coefficients are 0.05 for candidate 1 and 0.2 for each incorrect candidate. These are coefficients before gating, conflict projection, and the norm cap; they do not measure the eventual KD gradient magnitude. The reference CE coefficient is $0.1 \times 0.25 = 0.025$ because the group contains a correct response.

G.2 Why wrong does not imply negative advantage

Correctness and advantage serve different purposes. For a group with shaped rewards $(0.3, 0.1, 0, 0)$, all four responses can be mathematically wrong, yet the first has positive advantage. The all-wrong condition activates the full 0.1 reference CE coefficient, while relative policy learning can still distinguish these responses. Conversely, if all four rewards equal 0.3, every advantage is zero. In that case, both the policy gradient and admitted KD contribution vanish, leaving the reference objective active. If all four responses are correct and obtain reward 1.3, the same zero-advantage condition holds, but the CE coefficient is the smaller 0.025.

This distinction explains why fixing the KL direction does not make the ablation independent of answer correctness. Correctness still affects reward, the KD scale, gate features, and the reference fallback. Moreover, the historical fixed-reverse override can request reverse KD on an incorrect candidate with nonzero advantage; the full outcome rule does not do so. Neither statement implies that every requested KD contribution survives the local constraints.

G.3 What happens at one token

For a two-symbol illustration, let the student probabilities at a sampled multiplication symbol be $p = (0.7, 0.3)$ for $(+, \times)$, with positive advantage. Up to the positive factor A/T, the verifier-induced gradient is $(0.7, -0.7)$. Gradient descent decreases the addition logit and increases the multiplication logit. If an aligned teacher favors multiplication, its forward gradient can reinforce this direction. If its local gradient opposes the verifier-induced direction, the projection removes the opposing component before the gate and norm cap determine the admitted teacher update.

This illustration assumes a shared two-symbol probability space only to explain the gradient signs. In the actual experiment, the teacher and student do not share a tokenizer; their comparable quantities are constructed through the window, grouping, and vocabulary mapping described in Appendix A. The verifier does not generate the teacher's probability distribution, and the teacher is not guaranteed to repair every mistaken student prefix.

H Direction, probability mass, and the injected update

H.1 Restricted matching differs from full-vocabulary KL

The teacher and student assign probability mass both to matched and unmatched tokens. A direction comparison must retain that distinction. Consider a single aligned position with a one-to-one mapping, and write q_j for the teacher probability mapped to matched student coordinate j. Let M be the matched student coordinates, $p = \mathrm{softmax}(z)$, and

$$m_q = \sum_{j \in M} q_j, \qquad m_p = \sum_{j \in M} p_j.$$

For the restricted forward term,

$$D_F = \sum_{j \in M} q_j \log \frac{q_j}{p_j}, \qquad \frac{\partial D_F}{\partial z_k} = m_q p_k - \mathbb{I}[k \in M] q_k. \tag{16}$$

This follows from $\partial \log p_j / \partial z_k = \mathbb{I}[j = k] - p_k$ and a fixed teacher. In particular, an unmatched student logit can receive a gradient through softmax normalization even though its token is absent from the matched sum. If M covers the full vocabulary and $m_q = 1$, the familiar $p - q$ gradient is recovered.

For the restricted reverse term, define

$$D_R = \sum_{j \in M} p_j \log \frac{p_j}{q_j}, \qquad B = \sum_{j \in M} p_j \left(\log \frac{p_j}{q_j} + 1 \right) = D_R + m_p.$$

Then

$$\frac{\partial D_R}{\partial z_k} = p_k \left[\mathbb{I}[k \in M] \left(\log \frac{p_k}{q_k} + 1 \right) - B \right], \tag{17}$$

where the logarithm is evaluated only for matched coordinates. The derivation assumes positive probabilities; the executable loss uses numerical stabilization. These equations characterize the matched contribution before the unmatched L_1 term, alignment chain rule, and local gradient constraints are applied.

Restricted terms need not be nonnegative. For example, if the only matched pair has $q_j = 0.1$ and $p_j = 0.5$, its forward contribution is $0.1 \log(0.2) < 0$. This does not contradict KL nonnegativity, whose normalized-distribution assumptions do not hold for that isolated sub-sum. Renormalizing matched probabilities would define a different training objective. Similarly, sorting unmatched probabilities compares their ordered masses rather than identifying semantically equivalent tokens. Consequently, generic "mode covering versus mode seeking" intuitions do not by themselves determine the behavior of the implemented hybrid loss.

H.2 Alignment merging and gradients

Away from probability floors, normalized multiplication of several distributions has a simple logit interpretation:

$$\frac{\prod_{t\in G}\operatorname{softmax}(z_t)_v}{\sum_w \prod_{t\in G}\operatorname{softmax}(z_t)_w} = \operatorname{softmax}\left(\sum_{t\in G} z_t\right)_v. \tag{18}$$

The individual softmax partition functions cancel. Thus an aligned-group loss backpropagates through the merged representation into its member positions. This identity explains the gradient path; it does not make the merged vector an exact probability distribution over cross-tokenizer text spans. Clamps and the handling of singleton groups are part of the executable implementation and should be retained when reproducing it.

H.3 Projection and norm control have distinct functions

For fixed nonnegative gate and scale, set $u = \lambda ehd$ at one position, with $r \neq 0$. The ideal rule can also be understood as projecting u onto

$$\mathcal{C}_r = \{k : \langle r, k\rangle \geq 0,\ \|k\|_2 \leq \rho\|r\|_2\}.$$

To see this, first project onto the half-space with normal r. If u lies outside it, decompose u into its component parallel to r and its orthogonal component. The negative parallel component is removed. If the resulting vector exceeds the norm bound, radial rescaling reaches the boundary of the ball. If u is already in the half-space, radial rescaling preserves that condition. In either case, this is the closest feasible vector in Euclidean distance. This is a characterization of the ideal local rule, not a new optimization theorem or a claim about the finite-precision positive-margin implementation.

A gate alone attenuates a conflict without necessarily removing it. Projection alone can retain an arbitrarily large orthogonal teacher component. The norm cap limits that remaining component relative to the policy gradient. The three operations play complementary mathematical roles within the jointly evaluated update.

The ideal rule also bounds the angle of the combined local gradient. Write $k = a\hat{r} + b$, where $\hat{r} = r/\|r\|$, $a \geq 0$, and $b \perp r$. Since $\|b\| \leq \rho\|r\|$,

$$\cos\angle(r, r+k) = \frac{\|r\| + a}{\sqrt{(\|r\|+a)^2 + \|b\|^2}} \geq \frac{1}{\sqrt{1+\rho^2}}.$$

For $\rho = 1$, the ideal combined logit gradient lies within 45 degrees of the verifier-induced gradient. This bound concerns a single local logit update; the parameter-space counterexample in Appendix B still applies, as do the numerical qualifications there.

H.4 Why the teacher vector is detached

The injected surrogate $\langle z, \operatorname{sg}(k)\rangle$ is used to obtain the intended first-order vector k. Without the stop-gradient, differentiating $\langle z, k(z)\rangle$ would add $(\partial k/\partial z)^T z$, including derivatives through projection, normalization, and gate-related quantities. That would implement a different update. The teacher-vector construction and gate training therefore remain separate from the student's backward pass. The surrogate's scalar value is not a conventional divergence value and should not be compared across runs as if it were the unmodified KD loss.

I What the direction ablation holds fixed

The ablation changes one configured rule, but its effects can propagate through training. Changing a candidate's matched-term direction changes its teacher gradient, which can change the gate's agreement target, projection, and cap. Subsequent student updates can then change future rollouts. Thus "the gate is held fixed" means that its architecture, optimization rule, and hyperparameters are shared, not that learned gate values or gradients remain numerically identical across variants. The comparison evaluates the direction policy within the full training system, not the effect of swapping a gradient while freezing all future trajectories.

Table 8: The three GALA direction policies. Entries give requested matched-term directions for eligible candidates under the benchmarked V6 recipe. Actual KD also requires nonzero advantage and survival of the gate, projection, and norm-control conditions.

Quantity	Fixed forward	Outcome	Fixed reverse
Correct candidate direction	Forward	Reverse	Reverse
Incorrect candidate direction	Forward	Forward	Reverse
Correct candidate base scale	0.05	0.05	0.05
Incorrect candidate base scale	0.20	0.20	0.20
Unmatched probability term	L_1	L_1	L_1
Verifier policy / gate / CE	Retained	Retained	Retained
Projection / norm cap	Retained	Retained	Retained
Reward / EOS criteria	Shared	Shared	Shared

Likewise, seed 42 specifies each run's initialization and random-number setup; it does not guarantee paired candidate strings throughout divergent optimization histories. Our final-score comparison is controlled at the training-configuration level. The reported comparison consists of the seed-42 runs specified in the experimental setting.

What SFT establishes. SFT uses the reference corpus but lacks the online teacher and group-relative objective. Its comparison with GALA measures the performance difference between complete training recipes. The supplemental same-support KD comparison in Table 2 instead retains these supervision sources while removing the control stack. Neither comparison supplies a teacher-free GRPO result under this protocol. We do not substitute measurements from different local training histories for that missing comparison.

J Execution, restoration, and evaluation accounting

J.1 Distinguishing prompts, candidates, and optimizer updates

With four student ranks, microbatch size one, and accumulation two, a data step accounts for $4 \times 1 \times 2 = 8$ prompts. Eight such steps form a full student-update cycle, giving 64 prompts and $64 \times 4 = 256$ candidate responses globally. Each rank consequently processes 16 prompts and 64 candidates per complete cycle. Candidate count four is a within-prompt group size, not the number of DDP ranks or the microbatch size.

For 4,757 data steps, $4,757 = 594 \times 8 + 5$. There are therefore 594 complete update cycles and a final partial cycle with 40 prompts globally. In Gemma's two-stage schedule, step 600 is an update boundary: $600 = 75 \times 8$. Continuation covers another 4,157 data steps, or 519 complete cycles and one partial cycle, for 595 total optimizer updates. These calculations assume the stated nominal schedule and do not replace measurements of accepted examples or generated tokens.

The 38,056 nominal prompt occurrences exceed the 30,441 distinct training examples, so prompt occurrences must not be reported as additional unique training problems. Likewise, 152,224 nominal candidates do not imply that the same number of candidates receive KD: zero advantages, eligibility conditions, and local constraints can suppress that branch. The 1,024-token generation cap is an upper bound per candidate, not the average rollout length.

J.2 Prefetch and restoration boundaries

Prefetching eight data steps is consistent with the eight-step student-update interval only if generated responses remain within one frozen student-parameter cycle. A queue that crosses the parameter update would change the sampling history. Cached tokenizer mappings and fixed vocabulary indices can be reused across updates, whereas sampled responses and teacher scores associated with particular responses must remain attached to their originating prompt and student cycle.

A weight-only checkpoint supports inference, but does not define an exact training continuation. For the CPU FP32-master optimizer, the BF16 student tensor alone cannot reconstruct the higher-

precision state used for future updates. Table 9 lists the state required by the stated continuation semantics. It is a reproducibility specification, not a claim that the paper source archive contains all model checkpoints.

Table 9: State needed to interpret or reproduce a continuation. Checkpoints taken away from an update boundary additionally require accumulated gradients and the partial-cycle state.

State	Why it matters
Student weights and tokenizer	Define the conditional model, chat formatting, and ending-token interpretation.
FP32 masters and optimizer moments	Preserve accumulated parameter precision and adaptive update history.
Scheduler state and horizon	Distinguish the 75-update initial horizon from the 595-update continuation configuration.
Gate weights and optimizer state	Preserve the learned teacher-gradient attenuation mechanism.
Data cursor and sampler state	Prevent continuation from restarting at the beginning of the training stream.
Per-rank RNG states	Preserve each rank's sampling and stochastic-operation state as far as the execution backend allows.
Step and accumulation counters	Locate the next optimizer boundary and distinguish data steps from parameter updates.

A seed and a saved state do not guarantee bitwise identity across GPU architectures, reduction orders, kernels, or inference-library versions. The matched experiment protocol fixes the intended training variables; backend details and source hashes remain separate reproducibility artifacts. No throughput or wall-clock advantage is inferred from the use of H200 hardware.

J.3 Test aggregation and distinct denominators

The final test statistic is one greedy response per problem:

$$\widehat{\mathrm{Acc}}_{\mathrm{raw}} = \frac{1}{10,000} \sum_{j=1}^{10,000} c_{\mathrm{raw},j}.$$

It is not best-of-four or pass@4. Training's four sampled candidates are used for relative learning, not for selecting the best test response. Sharded evaluation should concatenate results by unique problem index, reject duplicate or missing indices, and aggregate correct counts over the global 10,000-item denominator; averaging shard percentages is only equivalent when shard sizes match. Prompt token hashes identify formatting or truncation differences that a matching problem index alone cannot detect.

The late-training table instead contains sampled candidate-level metrics from steps 4,258–4,757 under a 1,024-token cap. Greedy final evaluation uses a 2,048-token cap. These statistics differ in sampling policy, model checkpoint coverage, and response budget. A high sampled candidate correctness rate is not an estimate of the final greedy test accuracy, and neither quantity can be converted to best-of-four accuracy without additional joint information.

K Completion auditing: cases and metric definitions

The distinction between mathematical correctness and termination is operational rather than semantic: a valid arithmetic expression can occur inside an output that is unsuitable as a completed answer. Table 10 gives constructed examples for the strict audit in Appendix E. These are specification examples, not measurements of how often each case occurs in the reported runs.

An ending event must be established from actual generated token IDs and trusted termination metadata for the relevant tokenizer. A closed answer tag alone does not establish termination. A response

Table 10: Illustrative completion-audit cases. "Tail" refers to characters after the closing answer tag, inspected before stripping whitespace. The proposed penalty is part of a later recipe, not the reward used to obtain the paper's accuracy table.

Case	Raw correct	Natural end	Valid form	Strict correct
Correct answer, normal EOS, no tail	1	1	1	1
Correct answer, EOS, 8 spaces after tag	1	1	1	1
Correct answer, EOS, 9 spaces after tag	1	1	0	0
Correct answer, EOS, 1,626 newlines	1	1	0	0
Correct answer, clean tag, length cutoff	1	0	1	0
Empty response followed by EOS	0	1	0	0
Wrong expression, valid form and EOS	0	1	1	0
Correct answer, EOS at token 1,024	1	1	1	1

whose final allowed token is a recognized EOS/EOT can end naturally, whereas reaching the cap without such evidence is a truncation. Synthetic EOS appended by postprocessing cannot be used to certify natural completion. Empty outputs, unclosed answer blocks, multiple answer blocks, and disallowed trailing content fail the terminal-format condition even when generation ends naturally.

For a separately specified strict recipe, the non-overlapping reward cases are

$$R_{\text{strict}}(y) = \begin{cases} -1, & \neg e_{\text{natural}}, \\ -1, & e_{\text{natural}} \wedge \neg f_{\text{terminal}}, \\ c_{\text{raw}} + 0.1f + 0.2u, & e_{\text{natural}} \wedge f_{\text{terminal}}. \end{cases} \tag{19}$$

The first two branches set the strict positive reward components to zero; they do not accumulate two penalties for the same candidate. Reverse KD would be admitted only for strict-correct candidates, with invalid or unfinished responses receiving zero KD but retaining their policy and reference-learning paths. This changes both rewards and candidate support. It therefore cannot be applied retroactively to explain the existing scores as if the repaired recipe had already been evaluated.

Given complete per-item records, raw and strict test accuracy would use the same denominator, and

$$\widehat{\text{Acc}}_{\text{raw}} - \widehat{\text{Acc}}_{\text{strict}} = \frac{1}{N} \sum_j c_{\text{raw},j}\, \mathbb{I}[\neg(e_{\text{natural},j} \wedge f_{\text{terminal},j})] \geq 0.$$

The difference counts mathematically correct outputs that fail completion auditing. Its value cannot be recovered from aggregate correctness and EOS rates alone because their per-response overlap is unknown. Empty-output rate, truncation rate, and illegal-tail rate should likewise have explicit denominators and may overlap; they should not be added as if they were disjoint categories.

The Gemma fixed-reverse diagnostics establish a mismatch between the measured arithmetic outcome and generation behavior. They do not identify a unique causal mechanism. Wrong-trajectory reverse KD, completion-insensitive rewards, tokenizer alignment, and their interaction remain distinct explanatory possibilities; no intervention isolating those possibilities is reported here.

L GOLD SCORE PROVENANCE AND PROTOCOL CORRESPONDENCE

The GOLD baseline in Table 1 uses the official report's Countdown scores (Patiño et al., 2025): $0.0305 \times 100 = 3.05\%$ for Gemma and $0.4235 \times 100 = 42.35\%$ for Llama. The values were read from the report's tokenizer-similarity table on September 25, 2026. The experiment owner confirmed that the training and evaluation settings are consistent with our main comparison. The machine-readable source record preserves the official source for the scores and the author's confirmation for protocol parity. No GOLD run was performed in this codebase during manuscript preparation. The confirmation establishes the intended configuration correspondence, not independently verified equivalence of token mapping, prompt token IDs, initialization, continuation state, software versions, or verifier execution. Those artifacts are needed for a direct reproduction, especially to explain the large Gemma difference.

Table 11: Settings whose correspondence was confirmed by the experiment owner for GOLD and GALA. This table records that confirmation, not an independently rerun implementation audit.

Comparison item	Author-confirmed setting
Teacher and students	Qwen3-4B-Instruct-2507; Gemma-3-1B-it and Llama-3.2-1B-Instruct.
Training and test data	The same Countdown revision, processing, prompts, and train/test split.
Training controls	The same seed, training budget, batching, rollout limits, optimizer settings, and model-specific continuation schedule.
Final checkpoint	Step 4,757, seed 42.
Final evaluation	10,000 problems; vLLM greedy decoding; 512-token prompt limit and 2,048-new-token output limit.
Verification and termination	The same mathematical correctness and EOS/EOT evaluation criteria.

Subtracting the baseline scores yields $44.87 - 3.05 = 41.82$ and $46.12 - 42.35 = 3.77$ percentage points for fixed forward, and $44.69 - 3.05 = 41.64$ and $45.80 - 42.35 = 3.45$ points for outcome conditioning. These are absolute percentage-point gains, not relative percentages or significance estimates.

The external GOLD comparison contextualizes the reported system scores. It does not substitute for a reproduction under one implementation. The three local GALA variants compare matched-term direction within a shared framework; gate, projection, the norm cap, and reference CE remain active. Their descriptive final ranking and single-seed scope are stated in Section 4.

M CONNECTIONS BETWEEN THE SUPERVISION SOURCES

Targets, histories, and distributions. Sequence-level distillation supplies teacher-produced sequences as training targets (Kim & Rush, 2016). Dataset aggregation studies learning on the histories induced by a learner's own decisions (Ross et al., 2011); GKD applies student-generated sequences to language-model distribution matching (Agarwal et al., 2024). These viewpoints clarify the two teacher-related paths in GALA: offline reference CE supplies a target sequence, while online teacher scoring supplies distributions conditioned on sampled student continuations. The experiment uses both paths, with different models providing the offline reference corpus and online scores as stated in Section 4.

Comparable outputs and compatible gradients. ULD and approximate likelihood matching address the representation mismatch between teacher and student tokenizations (Boizard et al., 2024; Minixhofer et al., 2026). GOLD and X-Token further develop aligned or projected cross-tokenizer distribution comparisons (Patiño et al., 2025; Sreenivas et al., 2026). GALA treats comparability as the input to a second operation: deciding how the resulting teacher gradient enters reward-based learning. PCGrad and CAGrad provide related approaches to interacting objectives (Yu et al., 2020; Liu et al., 2021); the present construction makes the verifier-induced logit gradient the reference for an asymmetric projection and magnitude bound.

Verification as a specified interface. Answer verification and process supervision provide feedback at different granularities (Cobbe et al., 2021; Lightman et al., 2024). The executable Countdown verifier checks a final expression, and its group-relative advantage supplies a sequence-wide policy signal. Completion auditing adds an explicit interface requirement to mathematical validity. The proxy-reward perspective of Skalse et al. (2022) motivates distinguishing these observable properties, while Tables 3 and 10 provide the experiment-specific diagnostics and constructed audit cases.

M.1 Objectives, generated data, and feedback granularity

How an objective depends on its training data. DistiLLM combines a skew-divergence loss with adaptive off-policy reuse of student-generated outputs (Ko et al., 2024). DistiLLM-2 uses a contrastive construction that treats teacher and student responses differently (Ko et al., 2025). These results motivate separating the divergence choice from the distribution of sampled trajectories. The GALA direction ablation holds the rollout and remaining training recipe fixed, making the orientation of the matched-vocabulary contribution the variable of interest. Its correctness-dependent KD coefficient and verifier-induced advantage remain separate parts of the update.

Representation matching and offline transfer. DSKD unifies output spaces and introduces cross-model attention to align representations across models with different vocabularies (Zhang et al., 2024). MiniPLM transfers teacher information by refining the pre-training corpus through offline scoring (Gu et al., 2024b). These approaches organize teacher information at different points in the learning pipeline. GALA scores sampled continuations online, constructs an aligned hybrid loss, and controls its local gradient. This distinction identifies the role of each mechanism without assigning the performance of one experimental setting to another.

Supervision can take different forms. Distilling Step-by-Step uses teacher rationales as additional supervision for smaller models (Hsieh et al., 2023). Comparisons of process- and outcome-based feedback distinguish final-answer errors from errors in intermediate reasoning (Uesato et al., 2022). InstructGPT combines supervised demonstrations with reinforcement learning from a learned human-preference reward (Ouyang et al., 2022). In GALA, the feedback source is an executable arithmetic check; the teacher contributes conditional token distributions, while reference CE supplies an additional target. These distinct roles explain why sequence-level mathematical feedback is combined with distribution supervision rather than interpreted as a set of independently verified token labels.

Gradient scale and trade-offs. The multi-objective formulation of Sener & Koltun (2018) seeks a Pareto solution across tasks, while GradNorm adapts gradient magnitudes to balance training (Chen et al., 2018). GALA instead assigns a specific reference role to the verifier-induced gradient: projection removes an opposing component from the teacher gradient, and the relative cap bounds its admitted magnitude. The learned gate then provides attenuation within that explicit construction. These related optimization perspectives clarify which aspects are controlled by geometry, scale, and learned diagnostics.

Recent alignment and verifier-guided alternatives. BLD transfers teacher predictions through a common byte interface (Singh et al., 2026); CDM uses context to improve both sequence alignment and vocabulary mapping (Chen et al., 2025); SeDi combines tokenization-level semantic alignment with vocabulary re-encoding and distribution-aware supervision (Wang et al., 2026). These are alternative alignment mechanisms, not evidence that GALA's inherited mapping is competitive with them. A controlled comparison would replace the alignment path while holding the verifier, reference anchor, and gradient-control recipe fixed, and would separately report coverage and runtime.

CTPD combines shared character spans with preference distillation (Nguyen et al., 2026). VERPO uses evidence-conditioned self-distillation and a stopped controller that balances local reward alignment against Fisher movement cost (Li et al., 2026). Both are closely related to combining distributional guidance with task feedback. GALA's specific construction uses a frozen cross-tokenizer teacher, the full aligned hybrid-loss gradient, Euclidean conflict projection, a learned agreement gate, and a verifier-relative norm bound. We do not claim that reward-guided distillation or local gradient compatibility is unique to GALA, or that these alternatives have been experimentally outperformed.

Lee et al. (2026) study how generating rationales toward a revealed answer can degrade verifiable-reasoning distillation despite correctness filtering. Their intervention concerns trajectory generation, whereas our online teacher scores a student's sampled trajectory. This distinction motivates stating the teacher's conditioning context explicitly; it does not demonstrate that GALA eliminates rationale-quality problems or that its offline reference corpus is immune to them.

N UNCERTAINTY, DIAGNOSTIC DEFINITIONS, AND CONTROLLED FOLLOW-UP DESIGN

The original aggregate counts support the conditional intervals below. Table 2 adds two completed component interventions with raw and strict scores. Alignment measurements, gate diagnostics, and remaining interventions are specified separately from those results.

N.1 WHAT UNCERTAINTY CAN BE COMPUTED FROM AGGREGATE COUNTS

For a fixed trained model, let C be its correct count on $N = 10,000$ test items and $\hat{p} = C/N$. Under an independent, representative test-item model, a 95% Wilson score interval is

$$\frac{\hat{p} + z^2/(2N)}{1 + z^2/N} \pm \frac{z\sqrt{\hat{p}(1-\hat{p})/N + z^2/(4N^2)}}{1 + z^2/N}, \qquad z = 1.959964.$$

Table 12 applies this calculation to the supplied counts, using `analyze_reported_uncertainty.py`. This is a sampling-model analysis of the aggregates, not a new evaluation. Accuracy on the fixed benchmark itself is an observed quantity. Interpreting the intervals beyond that benchmark requires the stated sampling assumptions; dependence between related problem instances can invalidate the independent-item interpretation. The intervals do not quantify training-seed variation or adjust for inspecting multiple policies and checkpoints.

Table 12: Conditional test-sampling uncertainty from supplied local counts. Intervals exclude training variation and selection effects; they are not paired comparisons. GOLD is excluded because its scores are external.

Student	Configuration	Raw accuracy (%)	95% Wilson interval
Gemma	SFT	30.08	[29.19, 30.99]
Gemma	GALA forward	44.87	[43.90, 45.85]
Gemma	GALA outcome	44.69	[43.72, 45.67]
Gemma	GALA reverse	24.24	[23.41, 25.09]
Llama	SFT	33.49	[32.57, 34.42]
Llama	GALA forward	46.12	[45.14, 47.10]
Llama	GALA outcome	45.80	[44.83, 46.78]
Llama	GALA reverse	43.71	[42.74, 44.68]

Why a paired interval cannot be recovered. For two models evaluated on the same problems, let n_{10} count items correct only for the first and n_{01} count items correct only for the second. Their accuracy difference is $(n_{10} - n_{01})/N$, but its uncertainty also depends on $n_{10} + n_{01}$. The two marginal accuracies determine the difference, not the discordance count. Consequently, overlapping marginal Wilson intervals neither establish equivalence nor constitute a paired significance test. The retained aggregates cannot support a paired bootstrap, McNemar test, or run-to-run standard deviation.

A prospective selection and replication protocol. A follow-up comparison should reserve a deterministic validation split from the training corpus before training, select direction and checkpoint on validation strict accuracy under the common decoding cap, and then freeze those choices. Because the current test set has already informed development, that split cannot retroactively make the existing comparison confirmatory; independent confirmation requires an untouched evaluation set. For each of at least three independent training seeds, all three directions should use the same data split and

initialization protocol. Report every seed's raw and strict accuracy, their mean and sample standard deviation, and per-seed paired differences on identical test items. Resampling must preserve problem identity across compared policies; seed-level variation should be reported separately rather than treating seed–item pairs as independent observations. Three seeds are a minimum replication target, not a guarantee of sufficient power for sub-percentage-point differences.

N.2 Completed component controls and remaining interventions

The same-support unmodified-KD and fixed-gate controls are now reported in Table 2. The first retains the verifier, alignment, KD coefficient, eligibility rule, and CE while removing projection, gating, and capping together. The second sets $h = 1$ while retaining projection and the cap. These controls test the full transformation and the gate's incremental contribution, respectively. Table 13 also lists interventions that remain unevaluated.

Table 13: Component-control definitions. Same-support unmodified KD and the fixed gate are reported in Table 2; all other rows remain proposed interventions, not completed results.

Intervention	Definition and comparison purpose
Verifier + CE	Remove KD and gate training; retain group rewards and the reference anchor.
Aligned KD + CE	Remove the policy term and verifier-dependent KD controls; retain online rollouts and aligned forward KD with a fixed coefficient.
Unmodified KD (reported)	Retain the verifier and original eligibility rule; inject $k_{it} = \lambda_i e_i d_{it}$ without projection, gate, or cap.
Without projection	Set $\Pi_r(d) = d$; retain the learned gate and relative cap.
Without cap	Set $s_{it} = 1$; retain projection, gate, eligibility, and the small-gradient trust mask.
Fixed gate (reported)	Set $h_{it} = 1$; retain projection and cap.
Single-control variants	Starting from unmodified KD, enable only projection, only the cap, or only the gate.
Constant KD scale	Replace correctness-dependent λ_i with a validation-selected constant while retaining all controls.

For the KD-only control, removing the verifier loss while retaining its zero-gradient gate would trivially remove KD too. A meaningful verifier-free variant instead admits aligned nonempty trajectories without advantage filtering, uses a fixed KD coefficient, and fixes the CE coefficient. It is therefore a supervision-source comparison, distinct from an intervention on one gradient control. For the unmodified-KD control, retaining the original support isolates the vector transformation; a separate all-aligned-candidate control can measure the effect of reward-dependent eligibility.

All trained variants should share rollout *settings*; their generated trajectories will diverge as their parameters change. Complement training ablations with a frozen-checkpoint diagnostic that computes controlled and unmodified updates on identical cached prompt groups. If a norm-matched comparator is used, report that additional rescaling explicitly. To assess the inherited correct/incorrect KD ratio, compare the original $(0.05, 0.20)$ coefficients with equal coefficients selected on validation data, and report the resulting admitted gradient norms. Equal scalar weights do not ensure equal effective KD strength after gating and capping.

N.3 Alignment mapping and coverage diagnostics

Text-position grouping and vocabulary matching are separate objects. Greedy grouping may align one teacher token with several student tokens, or vice versa, while $\mathcal{M}$ indexes vocabulary pairs used at every aligned group. A matched pair list alone does not establish that each token index occurs at most once. The retained result records do not specify duplicate-normalization collision handling or certify injectivity of the executable map. The one-to-one assumption in Appendix H is therefore an assumption of that derivation, not a measured property of both tokenizer pairings.

An auditable map should include tokenizer revisions, normalization rules, special-token handling, the ordered pair list, and multiplicities on each side. Repeated normalized pieces require a documented collision rule. If a reproduction adopts deterministic one-to-one matching, it must mark that choice as a new implementation specification unless the original map is recovered and verified.

For any map, define its distinct matched supports $V_S^{\mathcal{M}}$ and $V_T^{\mathcal{M}}$. Coverage at aligned group a is

$$m_{S,a} = \sum_{v \in V_S^{\mathcal{M}}} p_a(v), \qquad m_{T,a} = \sum_{v \in V_T^{\mathcal{M}}} q_a(v).$$

Each coordinate is counted once here, so coverage stays in $[0, 1]$ even if the stored pair list contains duplicates. These support masses must be distinguished from pair-weighted sums used by a non-injective loss. Report vocabulary coverage as distinct matched IDs divided by vocabulary size, position coverage as aligned student response positions divided by attempted-window positions, and mass distributions separately for teacher and student. Include grouping-width quantiles, one-to-many/many-to-one text examples, suffix-fallback frequency over attempted candidates, and alignment failure rate over attempted windows. Stratify by student and correctness.

The unchanged sorted unmatched term can contribute substantially even when only matched-term direction is varied. Report matched and unmatched loss contributions and gradient norms alongside mass coverage. A window-size study should compare 32, 64, and 128 tokens with the remaining recipe held fixed and record the realized shared-text span, since equal token budgets need not cover equal text across tokenizers. These measurements are not recoverable from the supplied accuracy aggregates.

N.4 Activation, gate calibration, and parameter-gradient diagnostics

Report candidate-level and position-level quantities with separate denominators. For each correctness stratum, candidate summaries should include the nonzero-advantage fraction among sampled candidates and KD eligibility among candidates with nonzero advantage. Position summaries should include alignment coverage, the fraction suppressed by the small-$\|r\|$ trust rule among eligible aligned positions, conflict incidence before projection, and cap activation ($s_{it} < 1$) among nonzero pre-cap vectors. For eligible aligned positions, report gate mean, quantiles, and the fraction below a declared near-zero threshold such as 0.05. A zero denominator should be marked unavailable, not reported as zero activity.

Gate prediction should be evaluated against its pre-projection agreement target on held-out prompt groups, with class prevalence, binary cross entropy, Brier score, and a reliability diagram. Predicting an imbalanced agreement label accurately does not itself demonstrate a useful intervention. The reported $h = 1$ ablation shows only small incremental gains from the learned gate; its calibration and activation rates remain unmeasured. Gate outputs and cap activation should be reported jointly: when the cap is active, different positive gate values can produce the same final vector magnitude.

For frozen-checkpoint diagnostics, measure $\cos(r, d)$ and $\cos(r, k)$ over nonzero local vectors. Separately compute $g_V = \nabla_\theta \mathcal{L}_V$, $g_D = \nabla_\theta \mathcal{L}_D$, and $g_K = J^T k$, excluding CE for this diagnostic, and report $\cos(g_V, g_D)$ and $\cos(g_V, g_K)$ with consistent batch weighting. Include zero-vector rates rather than assigning them arbitrary cosines. Layerwise estimates or sketches must be identified as such. Proposition 1 concerns the first pair of measurements; it predicts neither the parameter-gradient cosines nor the realized effect of AdamW and CE. Any one-step reward comparison must clone the same checkpoint and optimizer state and evaluate both updates on a common held-out prompt set.

N.5 Joint completion-aware reporting at the final checkpoint

For every configuration in Table 1, a final audit should use one greedy response per test item at the same 2,048-new-token cap. Retain raw output text before whitespace stripping, token IDs, model-specific EOS/EOT IDs, trusted finish reason, prompt identity, and the verifier outputs. With c_j, e_j,

and f_j denoting raw correctness, natural ending, and terminal format as defined in Appendix E, report

$$\mathrm{Acc}_{\mathrm{raw}} = \frac{1}{N}\sum_j c_j, \qquad \mathrm{EndRate} = \frac{1}{N}\sum_j e_j,$$

$$\mathrm{FormatRate} = \frac{1}{N}\sum_j f_j, \qquad \mathrm{Acc}_{\mathrm{strict}} = \frac{1}{N}\sum_j c_j e_j f_j.$$

All four use the full test denominator; strict accuracy is not accuracy conditional on termination. Their joint records permit paired raw and strict comparisons without changing training rewards. Supplemental strict aggregates are now available for the three configurations in Table 2; per-item records and separate end/format rates were not supplied. Strict scores for SFT, external GOLD, outcome conditioning, and fixed reverse remain unavailable. Table 3 concerns sampled training candidates, so it cannot fill the missing final end or format rates. The reported Gemma reverse length-cap rate alone also does not identify the overlap of correctness, valid format, and genuine ending events. The raw–strict gaps in the component table are 5.30/5.80 points for unmodified KD, 3.80/4.40 for $h = 1$, and 3.87/4.32 for full GALA (Gemma/Llama). These differences count raw-correct answers rejected by strict evaluation; they do not identify which completion condition failed. Any later training repair must receive its own result row.