CABLE: CORRELATION-AWARE BARGAINING FOR MULTI-REWARD POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-reward post-training can maximize easy proxy rewards while task accuracy declines. We introduce CABLE (Correlation-Aware Bargaining with Liveness and Equalization), which combines correlation-based reward weighting with explicit control of normalization. It estimates interactions between objectives from token-logit correlations and combines their learning signals before a single actor backward pass. CABLE-L preserves reward-signal amplitudes; CABLE-LE adjusts weights using recent reward levels and adds entropy regularization. Across three training runs on Open-R1 coding with a 1.5B model, CABLE-LE reaches 12.56% LiveCodeBench-v6 pass@1 and a 36.37% four-suite average, improving on GAPO by 0.87 and 1.10 percentage points, respectively. On Math-M3, CABLE-L reaches a 52.21% five-benchmark average, compared with 51.32% for GRPO and 51.18% for the base policy. It also reduces mean response length from the base policy's 9,085 tokens to 3,203 tokens. On a matched actor-core workload, CABLE is over 4.2× faster than a parameter-gradient implementation of the same allocation rule.

1 INTRODUCTION

Language-model post-training combines rewards for correctness, execution or reasoning quality, and bounded response length (Shao et al., 2024; Liu et al., 2026b; Skywork AI, 2024). Their different scales, overlap, and learning dynamics make converting a reward vector into a productive policy update a central optimization problem.

Three quantities govern this update: *geometry*, whether objectives reinforce or oppose each other; *activity*, how broadly they supply reward contrast; and *attainment*, their achieved levels. Reward summation couples influence to scale and covariance (Shao et al., 2024). Separate normalization removes scale coupling (Liu et al., 2026b), but repeated standardization amplifies signals active on few prompts.

We introduce CABLE—*Correlation-Aware Bargaining with Liveness and Equalization*—to implement these controls (Figure 1). Token-logit correlations distinguish reinforcing, redundant, and opposing reward signals within the shared actor forward pass. A fixed number of coefficient updates turns these correlations into an allocation, motivated by bargaining among unit-norm objective directions rather than raw gradient magnitudes. Normalization controls signal amplitudes; optional level equalization gives more weight to objectives with lower recent rewards. The combined advantage drives one actor backward pass, without separate parameter gradients for every reward.

Across three coding training runs, CABLE-LE exceeds GAPO by 0.87 LiveCodeBench points and 1.10 four-suite average points. On Math-M3, CABLE-L reaches 52.21% average, exceeding GRPO and the base while shortening responses by 64.7%.

This paper makes three contributions:

- **Efficient correlation-aware allocation.** We estimate reward interactions from token logits, enabling coefficients to respond to redundancy and conflict rather than reward scale, without per-objective parameter backward passes.

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.

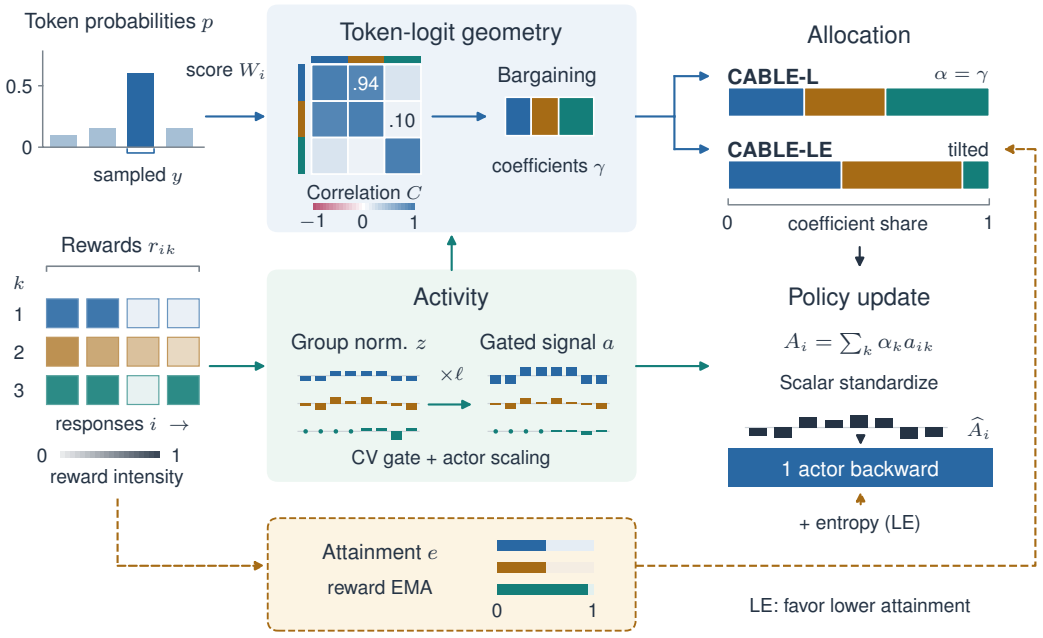


Figure 1: One CABLE update: reward signals and token scores determine correlation-based coefficients; CABLE-LE additionally applies an attainment tilt (gold branch) and entropy. Both combine signals before scalar standardization and one backward pass. Hue identifies objectives, intensity encodes reward, and strip width encodes allocation share. Signal miniatures illustrate $\eta = 0$; values are analytical examples.

- **Activity preservation and attainment equalization.** We characterize how normalization transforms objective activity and formulate reward-level equalization as a KL-regularized allocation with a closed-form solution.
- **Empirical gains and mechanism analysis.** We demonstrate coding and Math-M3 improvements, isolate attainment gains in coding ablations, and measure actor-side computational savings.

2 Related Work

Multi-reward policy optimization. GRPO standardizes summed rewards (Shao et al., 2024); GPG simplifies policy-gradient training (Chu et al., 2025). GDPO decouples normalization (Liu et al., 2026b), MO-GRPO modifies aggregation (Ichihara et al., 2025), GD^{2}PO filters conflicting rollouts (Liu et al., 2026a), and RDPO whitens advantages (Bai et al., 2026). CABLE separates interaction geometry, retained amplitude, and attained reward levels.

Reward activity and adaptive weighting. SAW adapts weights using reward CV (He et al., 2026); CABLE combines that activity statistic with correlations and explicit normalization order. Dynamic Reward Weighting uses hypervolume or accumulated gradient influence (Lu et al., 2025); CTWA uses reward–update covariances (Lu & Jiang, 2026); GradNorm and Dynamic Weight Averaging track gradient magnitudes or training rates (Chen et al., 2018; Liu et al., 2019). Our controller adjusts the geometric coefficients using exponential moving averages of reward levels, with a KL penalty.

Gradient geometry and bargaining. MGDA finds the minimum-norm convex combination of objective parameter gradients, yielding a common descent direction when that combination is nonzero; MGDA-UB uses shared-representation gradients to optimize an upper bound (Sener &

Koltun, 2018). PCGrad projects conflicting gradients (Yu et al., 2020), and CAGrad controls conflict relative to the average update (Liu et al., 2021). Nash-MTL bargains over objective parameter gradients (Navon et al., 2022); GAPO applies minimum-norm optimization to rescaled directions (Li et al., 2025). CABLE instead obtains correlations from token probabilities and detached advantages, applies coefficients directly to separately scaled actor signals, and uses one backward pass. It neither rescales coefficients by inverse gradient norms as Nash-MTL does nor optimizes GAPO's minimum-norm objective.

Preference-conditioned alignment. Rewarded Soups interpolates policies (Ramé et al., 2023); Rewards-in-Context, MODPO, and Directional Preference Alignment condition on requested preferences (Yang et al., 2024; Zhou et al., 2024; Wang et al., 2024). Projection Optimization handles nonlinear utilities (Xiong & Singh, 2025). CABLE instead adapts allocation within one online policy with fixed reward channels.

3 METHOD

CABLE combines token-logit allocation γ, reward-dispersion gates ℓ, and attainment EMA e into one scalar advantage. We index rollouts by n and synchronized actor microbatches by m.

CABLE-L retains gated amplitudes without attainment or entropy; CABLE-LE adds these controls and schedules objective scaling. Both standardize the combined scalar advantage.

3.1 PRESERVING REWARD-SIGNAL AMPLITUDES

For each prompt x, sample G responses from $\pi_{\theta_{\text{old}}}$ and evaluate M reward channels r_{ik}. With within-prompt mean μ_{xk} and population standard deviation σ_{xk}, form the decoupled signal (Liu et al., 2026b)

$$z_{ik} = \frac{r_{ik} - \mu_{xk}}{\sigma_{xk} + \epsilon}. \tag{1}$$

If reward k varies in a fraction f_k of equal-size prompt groups, ideal normalization gives $\mathrm{Var}(z_k) = f_k$. A second standardization amplifies its nonzero credit by $1/\sqrt{f_k}$ (Proposition 1); objective scaling therefore controls how sparse signals enter the update.

Following SAW's CV statistic (He et al., 2026), measure rollout-wide relative dispersion:

$$v_k^{(n)} = \frac{s_k^{(n)}}{|\bar{r}_k^{(n)}| + \epsilon}, \qquad \ell_k^{(n)} = \frac{v_k^{(n)}}{M^{-1}\sum_j v_j^{(n)}}, \tag{2}$$

Here $\bar{r}_k, s_k$ are raw reward means and population standard deviations. Gates have mean one; negligible mean CV gives unit gates. With signed transport $\mathcal{T}$, form actor-microbatch signals

$$b_{ik} = \mathcal{T}\left(\ell_k^{(n)} z_{ik}\right), \qquad a_{ik} = \frac{b_{ik} - \mu_{b,k}}{(\sigma_{b,k} + \epsilon)^{\eta_n}}. \tag{3}$$

Here b_{ik} is the activity-gated credit delivered to the actor; a_{ik} is the signal actually combined in its loss. The exponent η_n chooses whether to equalize each objective's variance ($\eta_n = 1$) or retain its amplitude ($\eta_n = 0$). Statistics are synchronized across workers. CABLE-L fixes $\eta_n = 0$; CABLE-LE transitions from one to zero on the schedule in Appendix B.

3.2 TOKEN-LOGIT INTERACTION GEOMETRY

Let ξ_{it} be the actor's temperature-scaled logits, $p_{it} = \mathrm{softmax}(\xi_{it})$, and y_{it} the sampled token. Its score in these coordinates is

$$h_{it} = \nabla_{\xi_{it}} \log p_{it,y_{it}} = e_{y_{it}} - p_{it}, \qquad \|h_{it}\|^2 = 1 - 2p_{it,y_{it}} + \sum_v p_{it,v}^2. \tag{4}$$

The squared norm measures how strongly the sampled token's log probability responds to a logit change: it approaches zero when that token already has probability one. All objectives share this

score and differ through a_{ik}. With response-token mask $m_{it} \in \{0, 1\}$, their surrogate directions and Gram are

$$u_k = \mathrm{concat}_{i,t}[m_{it} a_{ik} h_{it}], \qquad W_i = \sum_t m_{it} \|h_{it}\|^2,$$
$$S_{k\ell} = u_k^\top u_\ell = \sum_{i \in \mathcal{B}_{n,m}} W_i a_{ik} a_{i\ell}. \tag{5}$$

Thus $S = \mathbf{A}^\top \mathrm{diag}(W)\mathbf{A}$, with $\mathbf{A}_{ik} = a_{ik}$, weights reward interactions by token-score sensitivity within one shared forward pass. Appendix B relates it to parameter-gradient geometry.

3.3 Correlation-Based Allocation

Normalize the Gram to a cosine-similarity matrix with a diagonal floor,

$$\nu_k = \sqrt{\max\{S_{kk}, \epsilon_{\mathrm{alloc}}\}}, \qquad C_{k\ell} = \frac{S_{k\ell}}{\nu_k \nu_\ell}. \tag{6}$$

This removes individual signal norms: positive entries indicate reinforcement and negative entries indicate conflict. For three objectives, the fixed-step allocation map $\mathcal{B}_{80}(C)$ initializes $b^{(0)} = \mathbf{1}$ and applies 80 updates

$$b_k^{(j+1)} = \max\left\{ \epsilon_{\mathrm{alloc}}, \sqrt{\frac{b_k^{(j)}}{\max\{(Cb^{(j)})_k, \epsilon_{\mathrm{alloc}}\}}} \right\}, \qquad \gamma_k = \frac{b_k^{(80)}}{\sum_l b_l^{(80)}}. \tag{7}$$

Each coefficient responds to its current interaction $(Cb)_k$ with the combined direction, rather than only its own reward dispersion. We use $\epsilon_{\mathrm{alloc}} = 10^{-6}$ and uniform coefficients if any $C_{kk} \le \epsilon_{\mathrm{alloc}}$ or the coefficient sum is nonfinite. This fixed-step rule is distinct from the reference optimizer in Section 4.

Coefficients apply directly to a_{ik}. With inactive diagonal floors, positive objective rescaling cancels in C but remains in the combined advantage: geometry sets allocation, while Equation 3 sets amplitude.

3.4 Level Equalization as KL-Regularized Allocation

Fix reward bounds $L_k < U_k$ before training and map raw rollout means to utilities,

$$q_k^{(n)} = \mathrm{clip}_{[0,1]}\left(\frac{\bar{r}_k^{(n)} - L_k}{U_k - L_k} \right). \tag{8}$$

Initialize the attainment EMA from the first utility observation and update it as

$$e \leftarrow \beta e + (1 - \beta) q. \tag{9}$$

Actor updates share a detached rollout snapshot $e^{(n)}$ (timing: Appendix B.1). For CABLE-LE and $\tau > 0$, coefficients on the simplex Δ_M solve

$$\alpha^{(n,m)} = \arg\min_{\alpha \in \Delta_M} \left\{ \langle \alpha, e^{(n)} \rangle + \tau^{-1} \mathrm{KL}(\alpha \| \gamma^{(n,m)}) \right\}, \tag{10}$$

$$\alpha_k^{(n,m)} = \frac{\gamma_k^{(n,m)} \exp(-\tau e_k^{(n)})}{\sum_l \gamma_l^{(n,m)} \exp(-\tau e_l^{(n)})}. \tag{11}$$

The linear term favors lower attainment; KL anchors allocation to geometry, with τ controlling the trade-off. Equivalently, $\alpha_k \propto \gamma_k \exp(\tau d_k)$ for deficit $d_k = \max_l e_l - e_k$. Attainment is fixed per rollout; coefficients refresh per microbatch.

Algorithm 1 CABLE actor update

```
Set (η_n, τ, λ_H) = (0, 0, 0) for CABLE-L; use scheduled η_n and positive τ, λ_H for CABLE-LE.
for rollout n do
    Sample response groups; compute rewards, normalized credit z, and CV gates ℓ.
    Construct the detached attainment snapshot e^(n) for CABLE-LE.
    for synchronized actor microbatch B_{n,m} do
        Transport and scale signals a; score tokens and form W, S, C.
        Compute γ = B_80(C); set α = γ (L) or α = softmax(log γ − τ e^(n)) (LE).
        Combine and scalar-standardize credit; backpropagate Equation 13 once.
        Step the optimizer at gradient-accumulation boundaries.
    end for
end for
```

3.5 Single-Loss Policy Update

Set $\alpha = \gamma$ for CABLE-L and use Equation 11 for CABLE-LE. Combine objectives and standardize the scalar advantage:

$$A_i = \sum_k \alpha_k^{(n,m)} a_{ik}, \qquad \widehat{A}_i = \frac{A_i - \mu_A}{\sigma_A + \epsilon}. \tag{12}$$

Scalar statistics use the synchronized actor microbatch. Broadcast $\widehat{A}_i$ to response tokens and minimize

$$\mathcal{L}_{\mathrm{CABLE}} = \mathcal{L}_{\mathrm{clip}}(\widehat{A}) - \lambda_H \mathcal{R}[\mathcal{H}(p_{it})] + \lambda_{\mathrm{KL}} \mathcal{L}_{\mathrm{KL}}. \tag{13}$$

Here $\mathcal{L}_{\mathrm{clip}}$ is the clipped policy loss, $\mathcal{H}$ is entropy, and $\mathcal{R}$ reduces over response tokens; Appendix B specifies reductions and hyperparameters.

Allocation is detached. Algorithm 1 uses one backward per actor microbatch; matrix work costs $O(NM^2 + IM^2)$ for N responses and I iterations, beyond token-probability reductions.

4 Analysis of the Allocation Components

We characterize activity preservation, scale-invariant correlation geometry, and the attainment tilt. For the normalization identities, set $\epsilon = 0$ and map constant groups to zero. For positive-definite C, define the reference bargaining allocation by

$$b^* = \arg\min_{b>0} \left\{ \tfrac{1}{2} b^\top C b - \sum_k \log b_k \right\}, \qquad \gamma^* = \frac{b^*}{\mathbf{1}^\top b^*}. \tag{14}$$

The quadratic term measures the combined unit-direction norm, while the logarithms penalize neglecting any objective. At the unique minimizer, $b_k^*(Cb^*)_k = 1$ for every objective: its coefficient times its alignment with the combined direction is equalized, as in the bargaining formulation of Navon et al. (2022). This reference optimizer is distinct from the executed map $\mathcal{B}_{80}$. Proofs appear in Appendix A.

Proposition 1 (Activity retained by normalization)**.** *For equally sized complete prompt groups, let f_k be the fraction with nonconstant reward k. Then*

$$\mathrm{Var}(z_k) = f_k, \qquad \mathrm{Var}(\ell_k z_k) = \ell_k^2 f_k. \tag{15}$$

For $\ell_k > 0$ and $f_k > 0$, separately standardizing the gated signal produces $z_k/\sqrt{f_k}$. Centering before combination preserves the factors ℓ_k and $\sqrt{f_k}$.

Thus objective amplitude encodes both the gate and the fraction of prompts whose sampled responses receive different rewards. Dividing by the standard deviation raised to η leaves standard deviation $(\ell_k \sqrt{f_k})^{1-\eta}$; final scalar standardization controls the combined scale.

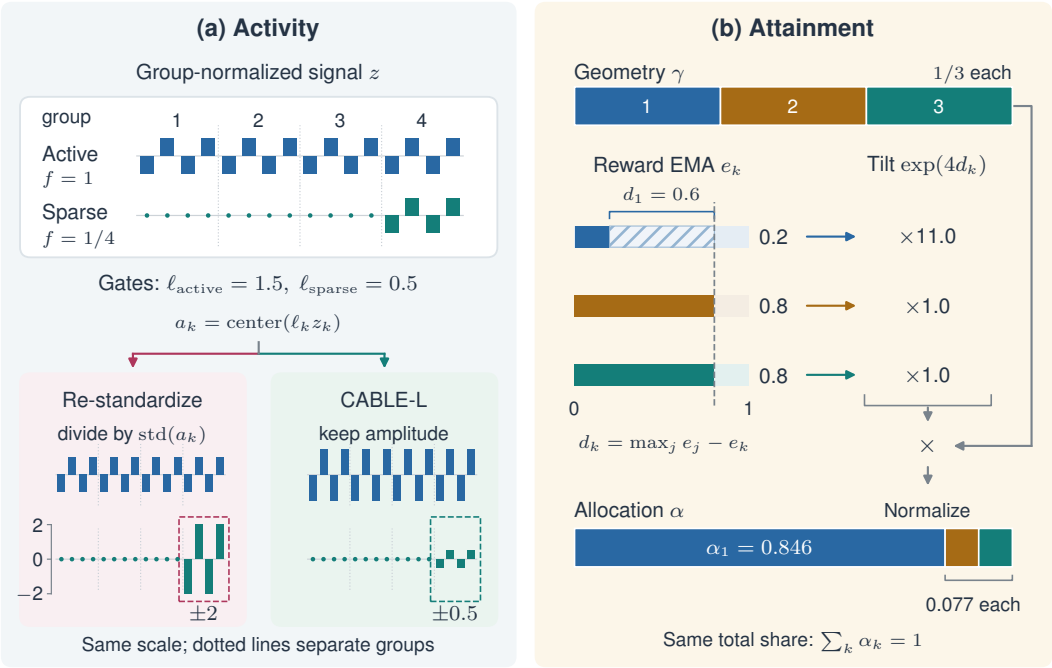


Figure 2: Analytical illustrations of two controls. **(a)** The same gated input undergoes per-objective standardization ($\eta = 1$) or amplitude preservation ($\eta = 0$), on a common amplitude scale. Standardization cancels positive gates and amplifies sparse credit. **(b)** At fixed geometry, lower attainment raises the blue share from $1/3$ to 0.846 ($\tau = 4$). Hatching marks attainment deficits; both coefficient strips sum to one.

With inactive diagonal floors, positive rescaling of nonzero directions leaves C unchanged, separating interaction geometry from activity (Proposition 4, Appendix A). For positive coefficients γ, attainment adds a separate control through

$$\frac{\alpha_i}{\alpha_j} = \frac{\gamma_i}{\gamma_j} \exp\big(\tau(e_j - e_i)\big). \tag{16}$$

At fixed geometry, decreasing e_i increases its relative allocation. Figure 2 illustrates how normalization preserves activity and how attained levels transform geometric coefficients. Appendix A.6 derives the KL allocation and its sensitivity.

5 Experiments

We evaluate code generation, mathematical reasoning, allocation components, and actor-side cost.

5.1 Experimental Setup

Tasks and metrics. We train Qwen2.5-Coder-1.5B-Instruct (Hui et al., 2024) on decontaminated Open-R1 Python problems (Open R1, 2025) with correctness, execution reliability, and length-validity rewards. Evaluation covers LiveCodeBench v6 (LCB; primary pass@1) (Jain et al., 2024), BigCodeBench-Hard (Zhuo et al., 2025), HumanEval+, and MBPP+ (Liu et al., 2023). We report their equal-weight pass@1 average and LCB pass@5.

For Math-M3, we train DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI et al., 2025) on DeepScaleR-Preview (Agentica, 2025) with correctness, frozen-Skywork PRM quality (minimum over reasoning steps), and length-validity rewards. We report sample-average accuracy on AIME24, AMC, MATH500, Minerva, and Olympiad, and their equal-weight mean.

Table 1: Code generation with a 1.5B policy (%). Trained rows report mean ± sample standard deviation across three seeds; the base is one fixed policy. Avg. equally weights four pass@1 suites. Bold and underline mark the best and second-best displayed means.

Method	LCB v6 pass@1	LCB v6 pass@5	BCB Hard	HumanEval+ pass@1	MBPP+ pass@1	Avg. pass@1
Base model	11.51	<u>16.81</u>	<u>6.08</u>	65.24	58.99	35.46
GRPO	9.79 ±0.84	12.25 ±1.18	5.86 ±1.03	62.60 ±0.93	59.52 ±0.79	34.44 ±0.41
GDPO	10.23 ±0.46	12.50 ±0.77	4.73 ±0.68	63.01 ±1.27	59.61 ±1.07	34.40 ±0.51
GPG	11.42 ±0.32	15.32 ±0.51	<u>6.08</u> ±0.68	<u>65.45</u> ±1.53	<u>59.79</u> ±0.26	<u>35.69</u> ±0.27
GD^{2}PO	9.43 ±0.50	11.59 ±0.64	**6.98** ±1.95	61.99 ±0.70	58.73 ±0.70	34.28 ±0.83
RDPO	9.99 ±0.42	12.24 ±0.61	5.41 ±0.68	64.43 ±0.93	<u>59.79</u> ±0.70	34.90 ±0.15
SAW	11.29 ±0.38	14.16 ±0.63	4.95 ±1.41	63.21 ±0.93	**59.96** ±0.15	34.86 ±0.55
GAPO	<u>11.69</u> ±0.07	**16.85** ±0.16	5.63 ±1.56	64.43 ±1.27	59.35 ±0.76	35.27 ±0.81
Dynamic Weighting	11.04 ±0.37	14.57 ±0.53	5.63 ±0.78	65.04 ±1.76	58.47 ±0.46	35.04 ±0.27
CABLE-L	10.92 ±0.60	13.66 ±1.09	4.28 ±0.78	64.43 ±1.86	59.61 ±0.15	34.81 ±0.66
CABLE-LE	**12.56** ±0.39	15.49 ±0.76	5.41 ±0.68	**68.09** ±3.57	59.44 ±0.15	**36.37** ±0.88

Comparison. Baselines include GRPO, GDPO, GPG, MO-GRPO, GD^{2}PO, RDPO, SAW, GAPO, and Dynamic Weighting. Methods share each task's initial policy, rewards, budget, and evaluator. GAPO and Dynamic Weighting use parameter gradients. Coding aggregates three training runs; Math-M3 uses one. Appendices B–C specify protocols and baseline-specific update settings.

5.2 Code Generation

CABLE-LE leads LCB pass@1, HumanEval+, and average in Table 1: $12.56 \pm 0.39\%$ LCB and $36.37 \pm 0.88\%$ average (sample SD). These improve on the base by 1.05 and 0.91 points. It exceeds GAPO at all three matched seeds, by 0.87 LCB and 1.10 average points overall, and exceeds GPG, the strongest three-seed average baseline, by 0.69 points.

5.3 Mathematical Reasoning

CABLE-L reaches 52.21% average (Table 2), exceeding GRPO by 0.89 points, the base by 1.03, and SAW by 1.53. Dynamic Weighting and GAPO reach 51.70% and 50.46%.

Accuracy and response length. At the same evaluation cap, CABLE-L shortens responses from 9,085 to 3,203 tokens while raising accuracy; length-target compliance rises from 37.49% to 75.09%.

Learning dynamics. Figure 3 reveals proxy takeover: between the first and final 20 rollouts, GDPO's mean correctness falls from 44.69% to 4.11% while process reward rises from 7.45% to 99.85%. CABLE-L instead raises correctness from 45.67% to 55.74% with 99.86% length validity.

5.4 Roles of Activity and Attainment

Task-dependent controls. On Math-M3, amplitude-preserving CABLE-L improves on geometry-only CABLE by 2.07 points and sustains correctness as length validity saturates. Coding benefits more from attainment calibration. Test correctness measures task success, but its reward is

Table 2: Mathematical reasoning with a 1.5B policy (%). Each trained method uses one seed and the common final endpoint. Avg. equally weights five benchmark accuracies. Bold and underline mark the best and second-best scores in this table.

Method	AIME24	AMC	MATH500	Minerva	Olympiad	Avg.
Base model	<u>32.71</u>	67.09	83.84	27.07	45.18	51.18
GRPO	29.58	68.37	**85.00**	**28.77**	44.88	51.32
GDPO	23.12	60.47	79.50	23.46	40.40	45.39
GPG	30.63	66.72	84.86	27.67	45.04	50.98
MO-GRPO	24.17	58.73	79.79	24.84	40.34	45.57
GD^{2}PO	17.50	58.66	78.71	23.39	39.14	43.48
RDPO	24.79	61.30	81.50	26.19	41.35	47.03
SAW	28.96	67.92	84.06	28.06	44.39	50.68
GAPO	29.58	65.89	83.99	28.10	44.72	50.46
Dynamic Weighting	31.46	**68.67**	<u>84.89</u>	27.90	<u>45.60</u>	<u>51.70</u>
CABLE-L	**33.33**	<u>68.60</u>	84.70	<u>28.58</u>	**45.82**	**52.21**
CABLE-LE	28.54	67.92	84.81	27.99	45.12	50.88

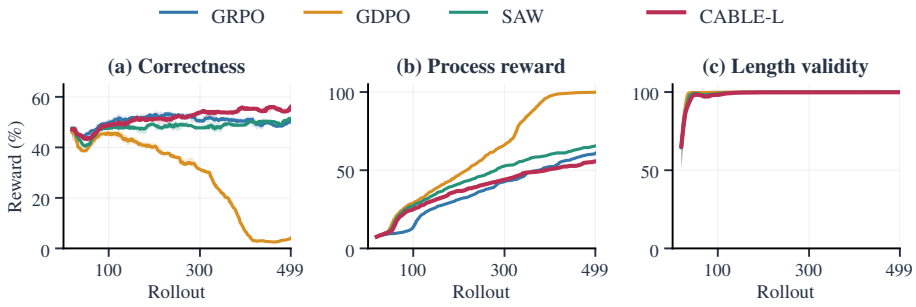


Figure 3: Math-M3: GDPO's process reward saturates as correctness collapses; CABLE-L sustains correctness gains. Lines: 20-rollout rolling medians; bands: within-window interquartile ranges.

lower than the execution and length rewards, so level equalization gives it more weight. Math's process probability instead measures a dense proxy. The coding factorial study verifies the attainment gain under both entropy settings.

Token-logit versus parameter-gradient geometry. Frozen-policy probes compare token-logit and parameter-gradient correlations. Math-base Pearson agreement is 0.991; all 37 pairs with true $|\cos| > 0.05$ have the correct sign, identifying reinforcement versus conflict. At the CABLE-L endpoint, agreement is 0.994. On Code, matching the estimator to the response-mean loss yields Pearson agreement of 0.999.

5.5 Computational Efficiency

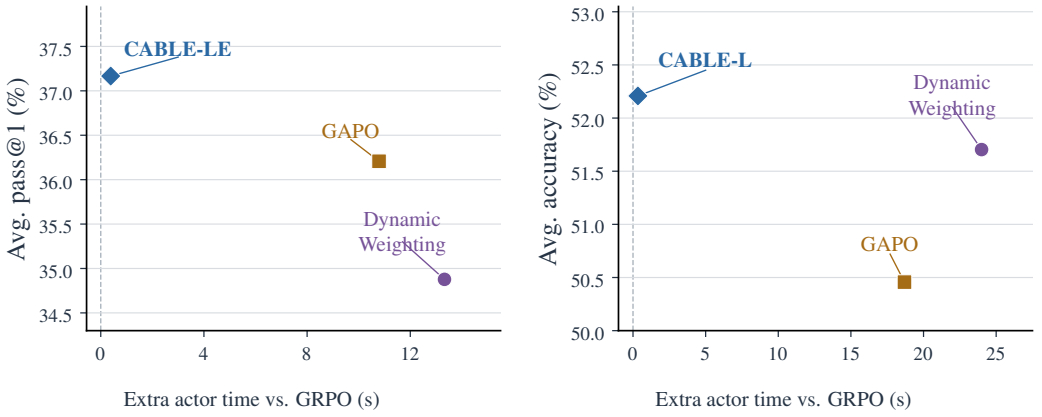
On a fixed Code-response workload, we compare each CABLE recipe with acquiring three full-parameter gradients for the same allocation rule. This isolates geometry acquisition, not the solver.

CABLE is 4.21–4.23× faster than matched parameter-gradient acquisition and within 8% of GRPO's actor-core time (Table 3). Peak memory is similar; the gain comes from avoiding repeated gradient acquisition.

Score–cost tradeoff. Figure 4 pairs actor-core time against GRPO: CABLE adds 0.39/0.34 seconds in Code/Math, versus GAPO's 10.79/18.69 and Dynamic Weighting's 13.33/24.00. Code uses original-policy scores. This measures interaction estimation, not end-to-end training.

Table 3: Actor-core cost: same 1.5B model, 256 Code responses, one B200, and allocation rule. Time is the median of ten 30-microbatch sweeps after three warmups; peak is allocated memory.

Actor route	Time (s)	× GRPO	Peak (GiB)
GRPO	5.32	1.00	56.28
CABLE-L	5.74	1.08	58.59
CABLE-LE (Code endpoint)	5.70	1.07	58.59
Matched parameter-gradient, L	24.14	4.54	58.57
Matched parameter-gradient, LE	24.13	4.54	58.57



(a) Code: four-suite average

(b) Math-M3: five-suite average

Figure 4: Accuracy versus extra actor-core time on task-specific frozen batches and one B200. Horizontal bars show IQRs of ten paired differences from GRPO. Code scores are original-policy observations, not three-seed means.

6 Conclusion

CABLE separates interaction geometry, activity, and attainment in one actor update. Token-logit allocation avoids per-objective parameter gradients; explicit scaling retains reward contrast; level equalization gives more weight to objectives with lower recent rewards. CABLE-LE improves coding across three training runs, while CABLE-L improves Math-M3 accuracy and shortens responses.

7 Limitations

Evidence and configuration selection. Evidence covers 1.5B policies: three coding runs and one Math run. Question-bootstrap intervals condition on original policies. GAPO leads replicated coding pass@5; Math CABLE-LE (50.88%) trails the base (51.18%). We do not establish scale transfer or task-independent configuration selection.

Mechanism attribution. Liveness jointly changes gating and normalization; isolated controls were not measured. Held-out Math profiles exclude PRM quality. Coding standardizes objectives, so its gains do not establish amplitude preservation. Reward levels need not represent comparable progress; CV mixes within- and between-prompt variation. Appendix B specifies clipping and controller timing. The finite allocation can be nonstationary and iteration-count sensitive under negative correlations; it is not a certified reference optimizer. Frozen probes do not establish deployment-wide fidelity.

Reproducibility Statement

Algorithm 1 and Appendices B–C specify the update, numerical safeguards, rewards, datasets, rollout budgets, decoding, endpoint selection, and training-seed aggregation. The main tables report completed endpoint evaluations. Appendix A provides the derivations, and Section 5.5 specifies the matched actor-core benchmark.

Statement on AI Assistance

AI assistants were used for implementation, experiment management, literature retrieval, and manuscript editing. The authors are responsible for the final verification of references, derivations, and empirical claims.

Ethics Statement

This work uses public tool-use, programming, and mathematics resources. Generated programs run with resource limits in dedicated training or evaluation environments. More capable code generation and tool use also increase misuse potential; deployment requires appropriate execution isolation, monitoring, and application-specific safety evaluation.

References

Agentica. DeepScaleR-Preview-Dataset. Dataset release, 2025. URL `https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset`.

Yang Bai, Kaiyuan Liu, Ziyuan Zhuang, Jiahong Zhou, Rongxiang Weng, Xin Chen, Jingang Wang, and Xunliang Cai. Multi-objective and mixed-reward reinforcement learning via reward-decorrelated policy optimization. *arXiv preprint arXiv:2605.13641*, 2026.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 794–803, 2018.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yuchen He, Baolong Bi, Shenghua Liu, Huaming Liao, Yuyao Ge, Bolin Wan, Siqian Tong, Juan Chen, Jiafeng Guo, and Xueqi Cheng. SAW: Stage-aware dynamic weighting for multi-objective reinforcement learning in large language models. *arXiv preprint arXiv:2606.07705*, 2026.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Mitsuki Sakamoto, Ryota Mitsuhashi, and Eiji Uchibe. MO-GRPO: Mitigating reward hacking of group relative policy optimization on multi-objective problems. *arXiv preprint arXiv:2509.22047*, 2025.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Chengao Li, Hanyu Zhang, Yunkun Xu, Hongyan Xue, Xiang Ao, and Qing He. Gradient-adaptive policy optimization: Towards multi-objective alignment of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 11214–11232, 2025.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, 2021.

Haotian Liu, Yihao Liu, Jingwei Ni, Siyuan Huang, Xinpeng Liu, Pengyu Cheng, Jiajun Song, Ruijin Ding, Junfeng Li, Zhechao Yu, Mengyu Zhou, Hongteng Xu, Xiaoxi Jiang, and Guanjun Jiang. GD^{2}PO: Mitigating multi-reward conflicts via group-dynamic reward-decoupled policy optimization. *arXiv preprint arXiv:2606.16771*, 2026a.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, 2023.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization. *arXiv preprint arXiv:2601.05242*, 2026b.

Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1871–1880, 2019.

Yining Lu and Meng Jiang. Uncovering cross-objective interference in multi-objective alignment. *arXiv preprint arXiv:2602.06869*, 2026.

Yining Lu, Zilong Wang, Shiyang Li, Xin Liu, Changlong Yu, Qingyu Yin, Zhan Shi, Zixuan Zhang, and Meng Jiang. Learning to optimize multi-objective alignment through dynamic reward weighting. *arXiv preprint arXiv:2509.11452*, 2025.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, 2022.

Open R1. Verifiable coding problems: Decontaminated, tested python training corpus. Dataset release, 2025. URL `https://huggingface.co/datasets/open-r1/verifiable-coding-problems-python_decontaminated-tested-shuffled`.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. Rewarded soups: Towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *arXiv preprint arXiv:2306.04488*, 2023.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, 2018.

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

Skywork AI. Skywork-o1-Open-PRM-Qwen-2.5-1.5B. Model release, 2024. URL `https://huggingface.co/Skywork/Skywork-o1-Open-PRM-Qwen-2.5-1.5B`.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*, 2024.

Nuoya Xiong and Aarti Singh. Projection optimization: A general framework for multi-objective and multi-group RLHF. *arXiv preprint arXiv:2502.15145*, 2025.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *International Conference on Machine Learning*, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025.

A PROOFS

A.1 SUMMED-REWARD NORMALIZATION

Proposition 2 (Implicit scale in a summed reward)**.** *For $R_i = \sum_k \lambda_k r_{ik}$ with $\lambda_k \geq 0$ and positive aggregate dispersion, ideal group normalization ($\epsilon = 0$) yields*

$$\frac{R_i - \mu_R}{\sigma_R} = \sum_k \frac{\lambda_k \sigma_{xk}}{\sigma_R} z_{ik}. \tag{17}$$

The coefficient on each standardized reward includes its within-prompt standard deviation.

Proof of Proposition 2. Linearity of the within-prompt mean gives $\mu_R = \sum_k \lambda_k \mu_{xk}$. Substituting $r_{ik} - \mu_{xk} = \sigma_{xk} z_{ik}$ yields Equation 17. In particular,

$$\sigma_R^2 = \sum_k \lambda_k^2 \sigma_{xk}^2 + 2 \sum_{k<\ell} \lambda_k \lambda_\ell \operatorname{Cov}_x(r_k, r_\ell). \tag{18}$$

If the standardized rewards are uncorrelated within the group, the contribution of reward k to the scalar advantage's variance is proportional to $\lambda_k^2 \sigma_{xk}^2$. Correlated rewards contribute the cross terms shown above. Zero-dispersion channels contribute zero centered signal; the identity applies to active channels when the aggregate dispersion is positive. □

A.2 REFERENCE BARGAINING SOLUTION

For positive-definite C, the Hessian of $f(b) = \frac{1}{2} b^\top C b - \sum_k \log b_k$ is $C + \operatorname{diag}(b_k^{-2})$, which is positive definite. The objective diverges when a coordinate approaches zero, and its quadratic term dominates the logarithms as $\|b\| \to \infty$. Thus it attains a unique minimizer in the positive orthant. Its first-order condition is $Cb = b^{-1}$.

After simplex normalization $\gamma^* = b^*/(\mathbf{1}^\top b^*)$, the condition becomes

$$(C\gamma^*)_k \gamma_k^* = \frac{1}{(\mathbf{1}^\top b^*)^2}, \tag{19}$$

which has the same value for every coordinate. Positive definiteness supplies the coercivity used in the existence proof. For a semidefinite matrix with a strictly positive null vector v, $f(tv) = -M \log t - \sum_k \log v_k \to -\infty$ as $t \to \infty$. Appendix B defines the finite numerical rule for observed batches.

A.3 TWO-OBJECTIVE REDUCTION

Proposition 3 (Two-objective reference allocation)**.** *Let $S = \left(\begin{smallmatrix} a & c \\ c & d \end{smallmatrix}\right)$ be positive definite. The positive solution of $Sb = b^{-1}$ has $b_2/b_1 = \sqrt{a/d}$. For its correlation-normalized matrix, $\gamma_1^* = \gamma_2^* = 1/2$.*

Proof of Proposition 3. The two equations are $ab_1 + cb_2 = b_1^{-1}$ and $cb_1 + db_2 = b_2^{-1}$. Multiplying by b_1 and b_2 gives $ab_1^2 + cb_1 b_2 = 1$ and $cb_1 b_2 + db_2^2 = 1$. Subtraction yields $ab_1^2 = db_2^2$. Positivity fixes the ratio $b_2/b_1 = \sqrt{a/d}$. A correlation matrix has $a = d = 1$, so its normalized weights are equal.

□

A.4 SCALE INVARIANCE AND OBJECTIVE CORRELATIONS

Proposition 4 (Scale-invariant reference allocation and three-objective geometry)**.** *For positive-definite S, let $D = \operatorname{diag}(S_{11}, \ldots, S_{MM})$ and $C = D^{-1/2} S D^{-1/2}$, without diagonal floors. Multiplying each surrogate direction u_k by a positive scalar leaves C, and hence γ^*, unchanged. For three objectives with*

$$C(c) = \begin{pmatrix} 1 & c & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad |c| < 1, \qquad \gamma_3^*(c) = \frac{\sqrt{1+c}}{2 + \sqrt{1+c}}, \tag{20}$$

the third objective's coefficient varies with the correlation of the first two.

Proof of Proposition 4. Let Q be a positive diagonal matrix rescaling the surrogate directions. The new Gram is $S' = QSQ$ and its diagonal matrix is $D' = Q^2 D$. Since diagonal matrices commute, $(D')^{-1/2} S' (D')^{-1/2} = D^{-1/2} S D^{-1/2} = C$. The unique ideal bargaining solution and its normalized coefficients are therefore unchanged.

For $C(c)$ in Equation 20, symmetry and uniqueness give $b_1 = b_2 = t$. The first equation gives $(1 + c)t^2 = 1$, while the third gives $b_3^2 = 1$. Hence $b = (1/\sqrt{1+c}, 1/\sqrt{1+c}, 1)$ and the claimed expression follows after normalization. Its derivative is positive for $|c| < 1$. □

A.5 Normalization and Reward Activity

Proof of Proposition 1. Let there be B equally sized prompt groups. For objective k, ideal group normalization has zero mean in each group and average squared value one in a nonconstant group, zero in a constant group. If Bf_k groups are nonconstant, pooling the groups gives

$$\mathrm{Var}(z_k) = \frac{1}{BG} \sum_{x,i} z_{ik}^2 = \frac{Bf_k G}{BG} = f_k. \tag{21}$$

Multiplication by a constant gate gives variance $\ell_k^2 f_k$. Its pooled standardization divides by $\ell_k \sqrt{f_k}$, leaving $z_k / \sqrt{f_k}$. Centering alone retains the gated amplitudes. □

With finite ϵ, the scaling follows the regularized second moment; unequal group sizes or actor microbatch sampling change the pooling weights.

For a strictly positive rollout-wide gate ℓ_k, independent centering and standardization cancel ℓ_k when $\epsilon = 0$. The same cancellation occurs in the ideal correlation matrix by Proposition 4. The $\eta = 0$ path in CABLE-L retains the gate by centering before aggregation and standardizing the combined scalar afterward. More generally, dividing by $(\ell_k \sqrt{f_k})^\eta$ gives standard deviation $(\ell_k \sqrt{f_k})^{1-\eta}$ when both factors are positive. These identities precede transport clipping and quantization.

A.6 KL-Regularized Attainment Allocation

For positive $\gamma \in \Delta_M$ and $\tau > 0$, the objective in Equation 10 is strictly convex on the simplex interior. Its Lagrangian stationarity condition is

$$e_k + \tau^{-1} \big(\log(\alpha_k / \gamma_k) + 1 \big) + \lambda = 0. \tag{22}$$

Exponentiation and simplex normalization give $\alpha_k = \gamma_k \exp(-\tau e_k)/Z$, where $Z = \sum_j \gamma_j \exp(-\tau e_j)$. This positive stationary point is the unique minimizer. Subtracting $\max_j e_j$ from every attainment coordinate leaves the normalized coefficients unchanged and recovers the deficit formula.

The attained objective value is $-\tau^{-1} \log Z$, and its gradient with respect to e is α. At fixed γ, the sensitivities are

$$\frac{\partial \alpha_i}{\partial e_i} = -\tau \alpha_i (1 - \alpha_i), \qquad \frac{\partial \alpha_i}{\partial e_j} = \tau \alpha_i \alpha_j \quad (i \neq j). \tag{23}$$

For utility EMAs in $[0, 1]$, Equation 16 bounds the multiplicative change in each coefficient ratio relative to γ between $e^{-\tau}$ and e^{τ}.

A.7 Reward Transport Bounds

For G centered observations x_i, $\sum_{j \neq i} x_j = -x_i$. Cauchy–Schwarz gives $\sum_{j \neq i} x_j^2 \geq x_i^2 / (G - 1)$, implying

$$\frac{|x_i|}{\sqrt{G^{-1} \sum_j x_j^2}} \leq \sqrt{G - 1}. \tag{24}$$

Adding a positive ϵ to the denominator preserves this bound. The mean-one nonnegative gate satisfies $0 \leq \ell_k \leq M$, so $|\ell_k z_{ik}| \leq M \sqrt{G - 1}$. The Math-M3 choice $M = 3, G = 16$ stays below 12; the ToolRL and Code group sizes give bounds below 8. These are sufficient transport ranges. The historical Open-R1 CABLE-LE branch uses the narrower range $[-3, 3]$ described in Appendix B, so the bound does not imply clipping-free transport for that cohort.

B Implementation Details

B.1 Scaling, Transport, and Controller Timing

Algorithm 1 operates on detached allocation statistics. CV gates use the complete rollout; objective and final scalar normalization use synchronized actor-microbatch statistics. CABLE-L fixes $\eta_n = 0$, while CABLE-LE uses

$$\eta_n = 1 - \mathrm{clip}_{[0,1]}\left(\frac{n-30}{20}\right). \tag{25}$$

The coding endpoint therefore uses per-objective standardization throughout; Math preserves amplitudes after rollout 50. Both configurations standardize the combined scalar advantage.

Controller timing. The attainment controller uses $\beta = 0.9$, $\tau = 4$, and reward ranges $[0, 1]$. It initializes from the first utility record and reconstructs the EMA in record order on resume. The Code reward writer indexes records from zero, whereas trained outer rollouts start at one: actor rollout n consumes records $0, \ldots, n-1$, including that rollout's reward mean. Calibration is thus rollout-level, not strictly past-only. Empty histories give unit multipliers; invalid or missing indexed records terminate the run. CABLE-LE uses entropy coefficient 2×10^{-4}; CABLE-L uses zero.

Transport and safeguards. Math liveness recipes use log-companded transport in $[-12, 12]$; Open-R1 CABLE-LE uses linear transport in $[-3, 3]$, clipping before actor normalization. Centering removes quantized-zero offsets. The allocation floor is 10^{-6}; negligible mean CV gives unit gates. Equation 7 executes 80 updates without a stationarity stopping test. Dead diagonals or nonfinite sums trigger uniform geometric coefficients, but finite oscillations do not. Normalizing coefficients does not certify reference optimality. Vocabulary-sharded probability reductions and data-parallel Gram sums are detached.

B.2 Component Recipes and Baselines

Geometry-only CABLE (development alias CorrNash) standardizes each objective, uses geometric coefficients, and omits CV gating, final scalar standardization, and entropy. Level-equalized variants add the rollout EMA tilt and the schedule above; their entropy arm adds the same coefficient as CABLE-LE. The full CABLE-LE additionally gates before transport and standardizes the scalar.

GRPO normalizes summed raw rewards; GDPO normalizes channels separately. GPG retains its single-update, unclipped, no-KL regime. MO-GRPO, GD^{2}PO, and RDPO retain their respective aggregation, conflict-filtering, and decorrelation rules.

Gradient-based baselines. GAPO acquires full actor gradients, solves the normalized-gradient minimum-norm problem with $p = 1$, and applies $\beta_k / \|g_k\|$ to the original directions. Dynamic Weighting sums microbatch token-mean probe losses, clips the full probe gradient to norm one, selects decoder layers 10–25, and updates weights once per rollout. Its new weights scalarize raw rewards before GRPO normalization. Released settings are initial weights $(0.334, 0.333, 0.333)$, influence scale 10^{-5}, learning rate 10^{-6} with polynomial power 1.03, AdamW betas $(0.9, 0.999)$, weight decay 0.01, zero entropy, KL coefficient 0.001, and clipping bounds 100.

Historical GAPO transfers minimum-norm coefficients without reciprocal norm rescaling; historical Dynamic Weighting uses surrogate row-sum influence. These historical implementations are distinct from the published-algorithm reproductions.

C Experimental Protocols

C.1 Training Tasks and Rewards

Code starts from Qwen2.5-Coder-1.5B-Instruct and trains on all 15,068 Open-R1 Python problems. Rewards measure the fraction of public input/output tests passed, execution without failure, and response length at most 1,536 tokens. Correctness compares complete normalized program output. Before training, all 13,134 tests on 80 audited gold solutions passed.

Math starts from DeepSeek-R1-Distill-Qwen-1.5B and trains on DeepScaleR-Preview. Rewards measure verified answer correctness, frozen Skywork PRM quality, and response length at most 4,000 tokens. The PRM scores nonempty newline-separated reasoning steps in deterministic windows of at most 4,096 PRM tokens; each step contributes once, and the response reward is the minimum step-quality probability. Empty or unscorable responses receive zero. All methods use the same frozen weights and validated Transformers Qwen2 implementation.

Table 4: Shared training settings. Budgets count rollout slots; Code slot zero initializes without an actor update. GPG retains its single-update regime; Dynamic Weighting retains the released settings in Appendix B.

Setting	Math-M3	Open-R1 Code
Rollout slots / endpoint	500 / 499	30 / 29
Prompts per rollout	512	128
Responses per prompt	16	8
Updates per rollout (clipped regime)	8	4
Maximum prompt tokens	1024	2048
Maximum response tokens (train)	8000	2048
Length-reward threshold	4000	1536
Actor learning rate	10^{-6}	10^{-6}
KL coefficient	0.0005	0.001
Training seeds	1234	4234, 5234, 6234
GPUs per training arm	8	8

C.2 External Evaluation

LiveCodeBench v6 covers 1,055 problems with ten candidates per problem, temperature 0.2, top-p 0.95, and a 2,000-token cap. We report pass@1 and pass@5. BigCodeBench-Hard uses 148 Instruct problems, greedy decoding, and a 1,280-token cap. HumanEval+ and MBPP+ use the expanded EvalPlus tests on 164 and 378 tasks, greedy decoding, and a 768-token cap. The coding average equally weights the four pass@1 suites.

All main coding policies use the common LCB scorer with four workers, identical per-program limits, and all failed outcomes retained. Runtime reconstructions passed fixed-candidate parity checks. The pinned BCB environment passes all 148 reference solutions; saved candidates are re-scored without regeneration. Component comparisons use their original common development scorer, explicitly separated from the main results.

Math evaluates AIME24, AMC, MATH500, Minerva, and Olympiad with sixteen responses per problem, temperature 0.6, top-p 0.95, and a 32,768-token cap. Suite accuracy is the fraction of all sampled responses answered correctly, not majority vote or best-of-16. The Math average equally weights suites. The PRM is absent from final answer evaluation.

Contamination checks. The Open-R1 audit found zero exact normalized-text overlaps with the four evaluation suites and zero shared Codeforces identifiers or URLs with LCB. A unigram/bigram TF–IDF scan found no train–test pair with cosine similarity at least 0.8; the maximum similarities were 0.408, 0.192, 0.486, and 0.358 for LCB, BCB-Hard, HumanEval+, and MBPP+, respectively. These checks address detected overlap, not unknowable pretraining exposure.

C.3 Endpoints and Aggregation

Every trained coding row uses the same three seeds and checkpoint 29. Means and sample SDs are computed across policies; each seed's equal-suite average is computed before its SD. Math uses seed 1234 and checkpoint 499. Intermediate monitoring never substitutes a method-specific best checkpoint. The complete Math checkpoint-199/399 grid was specified after the endpoint results and evaluated across all historical methods; it is a retrospective common-budget study.

Table 5: Measured coding component configurations at checkpoint 29, seed 4234 (%). BCB-Hard is uniformly re-scored in the reference-validated environment. This development comparison retains its original LCB scorer for every row; Table 1 instead uses the later uniform CPU4 rescore. The unchanged candidates explain the small LCB readout differences; scores from the two scorer versions are not used as a component contrast. Avg. equally weights the four pass@1 suites.

Method	LCB v6 pass@1	BCB Hard	HumanEval+ pass@1	MBPP+ pass@1	Avg. pass@1
Base model	11.44	6.08	65.24	58.99	35.44
CorrNash	9.44	4.73	62.80	59.26	34.06
CorrNash + entropy	10.87	5.41	65.85	59.79	35.48
Deficit-CorrNash	12.14	**6.76**	65.85	59.79	36.14
Historical Dynamic Weighting + entropy	8.81	4.73	64.63	58.73	34.22
GS-Entropy CorrNash	9.66	4.05	64.02	59.26	34.25
Active-set GS-Entropy CorrNash	9.40	3.38	64.63	59.79	34.30
CABLE-LE without activity recipe	**13.01**	5.41	67.68	**60.85**	36.74
CABLE-LE	12.93	5.41	**70.73**	59.52	**37.15**

Superseded evaluator results are excluded uniformly, rather than mixed with corrected scores. The main coding cohort and component study use different named scorers and do not form cross-scorer contrasts.

D Component Analysis and Measurement

D.1 Coding Components

Table 5 retains all measured configurations under the original common development scorer. The factorial comparison separates attainment and entropy; the full activity recipe also changes transport and scalar normalization. Original seed-4234 CABLE-L and CABLE-LE under the main evaluator obtain 10.86/13.00% LCB and 34.51/37.17% four-suite averages, respectively; these are not the three-seed mean contrasts.

D.2 Fixed-Policy Evaluation Uncertainty

We use 20,000 paired question-cluster bootstrap replicates, resampling independently within suites and preserving equal suite weights. Each question retains all ten Code candidates or sixteen Math responses. Code contains 1,745 questions and Math 1,560. These marginal percentile intervals condition on the original seed-4234 Code and seed-1234 Math policies; they are not intervals for the three-seed means or multiplicity-adjusted tests.

The original CABLE-LE policy's LCB gain over GAPO is 1.25 points, with interval $[0.07, 2.45]$. Math CABLE-L improves on the base by 1.03 points ($[-0.49, 2.56]$), GRPO by 0.89 ($[-0.32, 2.11]$), EMA SAW by 1.53 ($[0.43, 2.76]$), and geometry-only CABLE by 2.07 ($[0.67, 3.50]$). These fixed-policy comparisons and the replicated coding means describe different uncertainty sources.

D.3 Matched Actor-Core Measurement

The benchmark uses the same BF16 Qwen2.5-Coder-1.5B model, one B200, and 256 fixed Code responses: 223,547 prompt-plus-response tokens and 71,227 response tokens, packed into 30 micro-batches. Three warmup sweeps precede ten measured sweeps with rotating route order and synchronized CUDA timing. Every route includes the shared forward, allocation, and final backward. The matched parameter-gradient routes acquire three full gradients and use the same 80-step allocation as CABLE, isolating geometry acquisition rather than a different solver.

Table 6: Paired four-suite coding comparisons against the original seed 4234 CABLE-LE policy at the 1.5B checkpoint 29 endpoint. Differences and marginal 95% intervals are percentage points for those fixed policies, not three-seed mean contrasts. Each replicate resamples question clusters independently within suites, preserving equal suite weights and all candidates.

Comparator	**Avg. difference**	**Paired 95% interval**
Base	$+1.71$	$[+0.19, +3.27]$
GRPO	$+2.61$	$[+1.29, +4.02]$
GDPO	$+2.75$	$[+1.43, +4.10]$
GPG	$+1.39$	$[+0.23, +2.57]$
GD^{2}PO	$+2.13$	$[+0.68, +3.63]$
RDPO	$+2.12$	$[+0.67, +3.59]$
SAW (EMA)	$+1.13$	$[-0.16, +2.45]$
Historical GAPO	$+3.65$	$[+1.96, +5.36]$
Historical Dynamic Weighting	$+3.63$	$[+1.96, +5.31]$
SAW (stateless)	$+2.75$	$[+1.27, +4.31]$
GAPO	$+0.96$	$[-0.75, +2.67]$
Dynamic Weighting	$+2.29$	$[+0.78, +3.81]$

Median costs appear in Table 3; p95 times are 8.28, 8.53, and 8.78 seconds for GRPO, CABLE-L, and CABLE-LE, versus 34.75 and 35.54 for the matched L/LE parameter-gradient routes. These are sample percentiles, not confidence intervals. The workload is on-policy, so probability ratios are one and clipping is inactive. Optimizer steps, generation, rewards, distributed communication, checkpoint I/O, and controller reconstruction are excluded; the result is actor-core cost, not end-to-end throughput.