Consolidating Rewarded Perturbations for LLM Post-Training

Zheyu Zhang*, **Shuo Yang***, **Gjergji Kasneci**

Technical University of Munich
Munich Center for Machine Learning (MCML)
{name.surname}@tum.de

Abstract

Post-training of language models is commonly framed as a sample-score-update loop implemented by gradient descent. A recent line of work, exemplified by RandOpt, relocates this loop to weight space, sampling Gaussian perturbations around a pretrained model and ensembling the top-K rewarded specialists at inference. While competitive with PPO and GRPO under matched training compute, this prediction-level ensemble incurs K forward passes per test example and does not extend cleanly to free-form generation. We ask whether the rewarded population can instead be folded into a single deployable model, replacing the inference-time ensemble with one consolidated update. A split-half analysis over 25 model-task pairs reveals reproducible low-rank structure in every case. We turn this geometry into **CoRP** (**Co**nsolidating **R**ewarded **P**erturbations), a gradient-free operator that combines reward-weighted aggregation, compatibility-aware reweighting, and a held-out validation gate, with no gradient flowing through the language model. Across five language models from 0.5B to 8B and five tasks covering math, code, and creative writing, CoRP improves the base model by 8.1 points on average. Using one tenth of RandOpt's perturbation budget, CoRP exceeds single-inference RandOpt by 6.5 points and recovers more than half of the gain of the 50-pass majority-vote ensemble, at one forward pass per test example.

1 Introduction

Post-training is the alignment stage, where a broadly capable language model is adapted to produce responses preferred by humans while maintaining or improving performance, especially on reasoning-intensive tasks. Two families dominate current practice. *Reinforcement learning with verifiable rewards (RLVR)* [Ouyang et al., 2022, Schulman et al., 2017, Shao et al., 2024, Lambert et al., 2025] drives much of the recent progress on reasoning models [Guo et al., 2025, OpenAI et al., 2024, Team et al., 2025], and *on-policy distillation (OPD)* [Agarwal et al., 2024] trains the model on its own trajectories under a stronger teacher and now appears as a core stage in pipelines like Qwen3 and MiMo [Yang et al., 2025, Team et al., 2026]. Different as the two families look, both follow the same pattern: sample from the model, derive a per-trajectory or per-token learning signal, and update the weights to amplify capabilities that pretraining placed within reach.

A growing body of evidence suggests that what these updates accomplish is mainly to activate capabilities pretraining already made accessible. Chain-of-thought prompting elicits multi-step reasoning from models that were never trained on intermediate traces [Wei et al., 2022, Kojima et al., 2022]. RLVR continues to improve a base model even when its reward is partially incorrect or spurious, which motivates the hypothesis that the gradient update may operate beyond direct

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

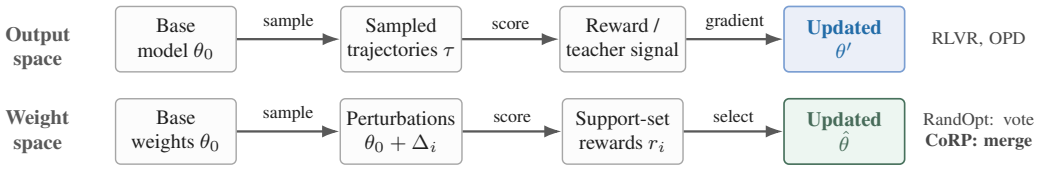


Figure 1: Two routes for self-elicitation post-training. Output-space methods sample trajectories and fold reward or teacher feedback back through gradients. Weight-space methods sample perturbations around θ_0 and fold rewarded variation back by aggregation. CoRP follows the weight-space route and returns a single deployable model instead of an inference-time ensemble.

reward maximization [Shao et al., 2026]. Mechanistic studies of reasoning fine-tuning reach a similar conclusion. They find that the update repurposes directions that are already active in the base model rather than installing new ones [Ward et al., 2025, Liu et al., 2025, Gandhi et al., 2025, Li et al., 2026a]. Taken together, these results suggest that post-training often exposes and stabilizes capabilities the pretraining step had already placed within reach.

As our baseline, Gan and Isola [2026] push this view from output space to weight space. The neighborhood of a contemporary pretrained model is dense with task specialists, and a single round of rewarded Gaussian perturbations is enough to expose them. RandOpt evaluates each perturbation on a small support set, keeps the top-K specialists, and votes their predictions at inference. The resulting ensemble matches or exceeds common post-training methods like PPO and GRPO at equal training compute, with no gradient through the language model. Where output-space recipes sample what the model can say, RandOpt samples what the model could be (see Figure 1).

This is effective at training but expensive at inference. Good performance needs $K \approx 50$, which means 50 forward passes per test example, and the prediction-level vote has to be redesigned for tasks whose outputs are not categorical. The natural fix is to consolidate the rewarded population into a single weight update. Reward-weighted averaging is the obvious first attempt, but rewarded perturbations behave as nearly orthogonal specialists, so naive averaging cancels rather than combines. Existing model-merging tools [Wortsman et al., 2022, Frankle et al., 2020, Ilharco et al., 2023, Yadav et al., 2023, Yu et al., 2024, Matena and Raffel, 2022] assume independently fine-tuned checkpoints. Since random rewarded perturbations around a single base model are not fine-tuned checkpoints, their compatibility must be evaluated directly.

Our study begins with a measurement: Across 25 model-task pairs spanning five families and five tasks, a reward-driven random search around one base model concentrates its useful variance on a reproducible low-rank subspace in every single case we examine. A reproducible mean direction, by contrast, exists in only 40 percent of the cases. These results indicate the presence of shared structure, which is distributed across a family of compatible directions rather than a single broad mean.

Building on this, we introduce *Consolidating Rewarded Perturbations* (CoRP[1]), a gradient-free operator that turns a rewarded perturbation population into one deployable update. CoRP first forms a reward-weighted candidate direction. It then reweights each perturbation by its alignment with that direction and its orthogonal dispersion, suppressing high-reward perturbations that would inject incompatible structure. A held-out validation gate commits a candidate only when its gain is stable on examples that did not contribute to its construction [Stone, 1974, Lewis and Syrgkanis, 2021], and lets the method abstain when no candidate is reliable. Across five language models from 0.5B to 8B parameters and five tasks, CoRP uses 500 rewarded perturbations, one tenth of RandOpt's budget, and recovers more than half of its $K{=}50$ ensemble gain at one forward pass per test example.

2 Background

2.1 Rewarded Perturbations

Gan and Isola [2026] treat post-training as a problem of sampling and filtering in weight space. Let $\theta_0 \in \mathbb{R}^d$ denote the pretrained weights of a language model. Given a finite set of noise scales $\Sigma \subset \mathbb{R}_{>0}$ and a sample size N, RandOpt draws N Gaussian perturbations of θ_0,

$$\Delta_i \;=\; \sigma_i\, \xi_i, \qquad \xi_i \sim \mathcal{N}(0, I_d), \qquad \sigma_i \in \Sigma, \qquad i = 1, \ldots, N, \tag{1}$$

[1] Our code is available at https://github.com/oooranz/CoRP.

where each σ_i is sampled uniformly from Σ and each ξ_i is an independent isotropic Gaussian direction in $\mathbb{R}^d$. RandOpt then scores each perturbed model on a small support set D_{sup} of task examples,

$$r_i \;=\; \frac{1}{|D_{\text{sup}}|} \sum_{x \in D_{\text{sup}}} R(\theta_0 + \Delta_i;\; x), \tag{2}$$

where $R(\theta;\, x) \in \mathbb{R}$ is the task reward of the model with weights θ on input x. We refer to the resulting set $\mathcal{P} = \{(\Delta_i, r_i)\}_{i=1}^{N}$ a *rewarded perturbation population*. It is the basic object our method operates on.

2.2 Prediction-Level Ensembling

RandOpt predicts by retaining the top-K entries of a rewarded population and voting over the corresponding perturbed models,

$$E_K = \text{TopK}\big(\{r_i\}_{i=1}^{N}\big), \quad \hat{y}(x') = \text{MajVote}\big(\{f(\theta_0 + \Delta_i;\, x') : i \in E_K\}\big), \tag{3}$$

where $f(\theta;\, x)$ denotes the model output at input x under weights θ. The ensemble works because the neighborhood of a well-pretrained θ_0 contains many task-improving perturbations, and because those perturbations behave as specialists whose strengths increasingly diverge as scale grows [Gan and Isola, 2026]. The same two facts make a single update hard. It has to preserve what the rewarded population shares while ignoring directions that are individually strong but mutually incompatible.

3 Do Rewarded Perturbations Share a Common Structure?

A rewarded population is dense and diverse. Whether it also shares a reproducible structure is a separate question, and one the prediction-level ensemble does not need to answer. A pairwise test cannot answer it either. Independent Gaussian directions in a billion-dimensional parameter space are nearly orthogonal whether they are rewarded or not [Vershynin, 2018], so any shared signal has to be read at the population level. We read it from a split-half diagnostic in the spirit of classical reliability analysis [Brown, 1910, Spearman, 1910].

For each rewarded population we project the perturbations onto a fixed low-dimensional coordinate sketch $z_i \in \mathbb{R}^m$ with $m \ll d$. We rank the population by reward, partition the top-M indices into two disjoint halves S_A and S_B of equal size, and repeat the partition over 20 random splits. On each half we form a reward-weighted mean of the sketched perturbations, $\mu_A = \sum_{i \in S_A} w_i^{(1)} z_i$ and $\mu_B = \sum_{i \in S_B} w_i^{(1)} z_i$, with the reward-tilted weights $w_i^{(1)}$ defined in Eq. 7 below. The first statistic compares the two means,

$$C_{\text{mean}}(M) \;=\; \mathbb{E}_{\text{split}}\big[\cos(\mu_A,\, \mu_B)\big], \tag{4}$$

where the expectation is taken over random splits. The second statistic compares the top-r principal subspaces of the two weighted clouds, with the random-baseline overlap r/m between two random r-dimensional subspaces of $\mathbb{R}^m$ subtracted away,

$$C_{\text{sub-ex}}(M, r) \;=\; \mathbb{E}_{\text{split}}\Big[\, \tfrac{1}{r} \big\| U_A^\top U_B \big\|_F^2 \,\Big] \;-\; \tfrac{r}{m}, \tag{5}$$

where $U_A, U_B \in \mathbb{R}^{m \times r}$ are orthonormal bases of the top-r principal subspaces of the weighted point clouds on the two halves and r is fixed at a value reported in the appendix. We call a statistic *stable* when its nonparametric 95 percent lower confidence bound over the random splits lies above zero.

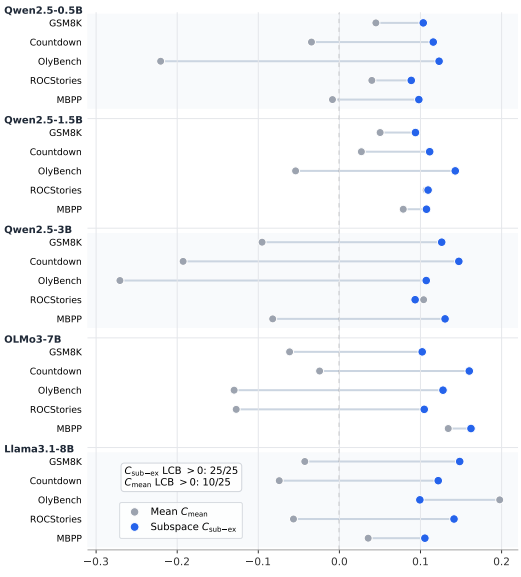


Figure 2: Split-half statistics at M=50. Each row is one model-task pair. Gray points are the lower confidence bound of the mean statistic C_{mean}, blue points the lower confidence bound of the subspace statistic $C_{\text{sub-ex}}$. The dashed line marks zero.

We run the diagnostic on 25 model-task pairs across five instruction-tuned models, Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, OLMo3-7B-Instruct, and Llama-3.1-8B-Instruct, and across five tasks covering arithmetic search (Countdown), grade-school math (GSM8K), olympiad-style reasoning (OlympiadBench), creative writing (ROCStories), and program synthesis (MBPP). Figure 2 reports the result. The subspace-excess statistic is positive with 95 percent confidence on 25 of 25 pairs. The mean-consensus statistic clears the same bar on only 10. Rewarded perturbations therefore share a reproducible signal in every case we examine, but in most cases that signal lives in a low-rank family of compatible directions rather than in any single mean direction.

Two consequences follow. First, reward-weighted averaging cannot be the answer on its own, because the reward-weighted mean is only reproducible on a minority of pairs. Second, the shared signal nonetheless exists in every pair, distributed across a low-rank family of directions rather than collapsed onto one. A consolidation operator therefore needs to do two things at once: pick a working direction from the rewarded population, and handle perturbations whose strengths lie outside it.

4 Consolidating Rewarded Perturbations

Reward-weighted averaging is the natural first attempt at consolidation, but it conflates usefulness with compatibility. A perturbation can earn a high reward on the support set and still point in a direction that the rest of the rewarded population does not share. Averaging then writes both the useful component of the perturbation and its incompatible component into the model, and the incompatible component is what cancels.

CoRP separates these two questions and answers them in sequence. A first pass uses reward alone to propose a direction. A second pass then asks which perturbations support the proposal and which add mass orthogonal to it, and reweights accordingly. A held-out validation gate commits the resulting candidate only when it continues to help on examples not used to construct it, and abstains when no candidate is reliable. No gradient flows through the language model at any step.

4.1 From Reward to Compatibility

Let $\mathcal{P} = \{(\Delta_i, r_i)\}_{i=1}^N$ be the rewarded perturbation population defined in Eqs. 1–2. CoRP builds a single weight update from this population by assigning weights to the sampled perturbations and normalizing the resulting direction.

The first pass retains the top-q reward quantile,

$$E_q \;=\; \{\, i : r_i \geq \tau_q \,\}, \qquad \text{where } \tau_q \text{ is the } q\text{-th quantile of } \{r_i\}_{i=1}^N, \tag{6}$$

and forms reward-tilted weights and a provisional direction,

$$w_i^{(1)} \;=\; \frac{\exp(\beta\, r_i)\, \mathbf{1}[i \in E_q]}{\sum_j \exp(\beta\, r_j)\, \mathbf{1}[j \in E_q]}, \qquad m^{(1)} \;=\; \sum_{i=1}^N w_i^{(1)}\, \Delta_i, \tag{7}$$

where $\beta > 0$ is an inverse temperature and $\mathbf{1}[\cdot]$ is the indicator function. The provisional direction $m^{(1)} \in \mathbb{R}^d$ is the reward-weighted mean of the elite perturbations. This first pass follows the weighted-sample logic used in reward-weighted regression, advantage-weighted regression, the cross-entropy method, and evolution strategies [Peters and Schaal, 2007, Peng et al., 2019, Rubinstein, 1999, Mannor et al., 2003, Salimans et al., 2017]. It is not yet a deployable update. Reward selects perturbations that help the task, but it does not say whether those perturbations can be written into the same model.

The second pass measures compatibility with the proposal. For each elite perturbation, CoRP computes its alignment with $m^{(1)}$ and its orthogonal dispersion against $m^{(1)}$,

$$a_i \;=\; \cos\!\big(\Delta_i,\, m^{(1)}\big), \qquad d_i \;=\; \big\|\Delta_i - \Pi_{m^{(1)}} \Delta_i\big\|^2, \tag{8}$$

where $\Pi_{m^{(1)}} \Delta_i$ is the projection of Δ_i onto the line spanned by $m^{(1)}$. The alignment $a_i \in [-1, 1]$ measures how strongly a perturbation supports the proposal, and the dispersion $d_i \geq 0$ measures how much of it lies outside. CoRP then combines reward and compatibility into the second-pass weights,

$$w_i^{(2)} \;=\; \frac{\exp(\beta\, r_i + \gamma_a\, z(a_i) - \gamma_d\, z(d_i))\;\mathbf{1}[i \in E_q]}{\sum_j \exp(\beta\, r_j + \gamma_a\, z(a_j) - \gamma_d\, z(d_j))\;\mathbf{1}[j \in E_q]}, \tag{9}$$

where $\gamma_a, \gamma_d \geq 0$ control the strength of the compatibility terms and $z(\cdot)$ standardizes a statistic to zero mean and unit variance over E_q. The consolidated direction and update are

$$\bar{\Delta} \;=\; \sum_{i=1}^{N} w_i^{(2)}\, \Delta_i, \qquad \hat{\theta} \;=\; \theta_0 + \eta\, \bar{\Delta} / \|\bar{\Delta}\|, \tag{10}$$

with deployed step size $\eta > 0$. The three terms in the exponent of Eq. 9 capture, in order, usefulness, support for the proposal, and resistance to orthogonal mass. The same principle of measuring compatibility before aggregation appears in parameter-space merging of independently fine-tuned checkpoints [Matena and Raffel, 2022, Yadav et al., 2023, Yu et al., 2024]. CoRP applies it to a different source family, random rewarded perturbations around one base model, where compatibility cannot be read from sign, magnitude, or curvature of fitted checkpoints and has to be measured against a population proposal.

4.2 Validating the Consolidated Update

The same support examples that weight the perturbations can also overstate the value of the resulting update. CoRP separates proposal generation from proposal validation, in the spirit of nested cross-validation and cross-fit estimation [Stone, 1974, Lewis and Syrgkanis, 2021]. We split D_{sup} into three disjoint folds A, B, and P. For each configuration in a small grid of (q, β, η), CoRP forms a candidate $\hat{\Delta}_{q,\beta}$ from rewards computed only on A, following Eqs. 6–10, and ranks candidates on A by a constructive score

$$s^A(q, \beta, \eta) \;=\; \frac{|F^A(q, \beta, \eta)| - \lambda\, |G^A(q, \beta, \eta)|}{|A|}, \tag{11}$$

where F^A is the set of examples in A that candidate $\theta_0 + \eta\, \hat{\Delta}_{q,\beta}$ answers correctly while base model θ_0 does not, G^A is the symmetric set of regressions, and $\lambda > 0$ penalizes regressions. A candidate passes the gate only if its constructive score on B is positive and the lower confidence bound of its accuracy change on B is positive. The probe set P then calibrates the deployed step size on a fixed multiplier grid $\{\alpha_j\}$, retaining the largest α_j such that $\theta_0 + \alpha_j\, \eta\, \hat{\Delta}_{q,\beta}$ has a positive lower confidence bound on ΔAcc_P. If no candidate passes the gate or no multiplier passes the probe, CoRP abstains and returns θ_0.

4.3 Iterating Around an Accepted Update

An accepted update changes the local neighborhood, and the rewarded population around the new center can carry useful structure that the original sampling did not reach. CoRP therefore iterates. The candidate construction, the held-out gate, and the probe calibration of the previous two subsections all remain in place. Only the sampling distribution changes between iterations.

Let θ_t denote the current center, with θ_0 the pretrained model and θ_1 the first accepted update. CoRP draws N_{loc} local perturbations

$$\Delta_j^{\text{loc}} \;\sim\; \mathcal{N}\big(0,\, \rho_t^2\, \Sigma_t\big), \qquad j = 1, \ldots, N_{\text{loc}}, \tag{12}$$

where $\rho_t > 0$ is the local search scale and Σ_t is a covariance whose low-rank component concentrates exploration in directions where the previous rewarded population varied, with an isotropic floor preserving coverage in directions the population did not emphasize. The exact form of Σ_t follows covariance-adaptive black-box search [Hansen, 2023, Maheswaranathan et al., 2019, Choromanski et al., 2019] and is given in the appendix. The covariance shapes the proposal distribution only and does not appear in the deployed update. The next center θ_{t+1} is committed only when the validation rule of §4.2 accepts a proposal on fresh A, B, P splits, and the iteration stops otherwise.

5 Experiments

We evaluate CoRP against RandOpt and gradient-based baselines on a shared benchmark and address four questions in turn. How much of the K=50 ensemble gain does a single CoRP model recover, and at what cost. When can a rewarded population actually be consolidated into one update. And what kinds of errors does the operator repair.

Table 1: Comparison to post-training baselines. We report test-set accuracy, except for MBPP where we report pass@1. **Blue** marks the overall best result when it is achieved by optimization-based methods. **Green** marks the best perturbation-based method.

Model	Method	**Countdown** (math)	**GSM8K** (math)	**OlyBench** (math)	**ROCStories** (writing)	**MBPP** (prog.)
Qwen2.5-0.5B-Inst	Base	0.1±0.1	39.9±0.0	4.2±0.4	22.0±0.0	30.9±0.0
	PPO	**14.8±2.8**	43.2±1.2	**16.1±0.5**	19.1±0.0	37.8±3.0
	GRPO	13.0±0.0	48.4±0.0	6.9±0.1	30.9±1.1	42.8±3.7
	RandOpt (K=1)	4.8±0.8	44.8±0.5	8.6±0.7	23.3±0.9	30.7±0.7
	RandOpt (K=50)	8.4±0.3	**54.1±0.8**	**15.8±0.7**	**32.2±0.4**	**46.2±0.4**
	CoRP	**14.5±1.2**	43.9±0.6	9.6±0.4	32.0±0.9	34.3±0.6
Qwen2.5-1.5B-Inst	Base	6.7±0.0	58.8±0.2	13.4±0.4	46.7±0.1	62.3±0.2
	PPO	27.0±0.0	71.6±0.7	26.3±0.1	51.8±0.8	69.5±0.4
	GRPO	27.5±0.2	72.1±0.7	18.8±0.8	**53.6±1.3**	**70.2±0.4**
	RandOpt (K=1)	9.1±0.3	59.1±0.8	19.9±0.7	46.6±0.2	61.1±0.6
	RandOpt (K=50)	**52.7±0.6**	**76.4±0.3**	**30.4±0.7**	48.5±0.7	**69.6±0.5**
	CoRP	26.2±0.7	70.1±0.3	20.5±0.5	**52.2±0.6**	64.0±0.5
Qwen2.5-3B-Inst	Base	10.0±0.1	79.8±0.4	24.5±0.2	54.7±0.1	69.5±0.3
	PPO	35.3±0.1	83.1±0.2	34.4±0.2	49.0±0.6	76.3±1.0
	GRPO	32.6±0.1	83.2±0.2	29.0±0.0	56.3±4.4	**77.0±0.9**
	RandOpt (K=1)	10.4±0.8	81.4±0.9	30.3±0.4	54.6±0.4	69.1±1.1
	RandOpt (K=50)	**58.4±0.2**	**87.1±0.6**	**39.2±0.6**	56.5±0.3	**75.9±0.6**
	CoRP	36.9±0.3	82.3±0.4	32.5±0.5	**59.1±0.6**	70.3±0.8
OLMo3-7B-Inst	Base	64.8±0.2	82.9±0.4	28.7±0.1	64.0±0.0	65.9±0.2
	PPO	69.0±0.0	88.4±0.4	28.0±0.2	64.7±0.6	67.7±0.6
	GRPO	68.5±0.7	87.0±0.2	27.9±0.6	**65.8±0.6**	70.8±2.2
	RandOpt (K=1)	74.4±0.7	83.2±0.6	37.5±0.3	64.2±0.8	66.5±0.6
	RandOpt (K=50)	**85.0±0.2**	**89.5±0.2**	35.4±0.4	64.5±0.3	**75.1±0.9**
	CoRP	78.9±0.4	84.8±0.1	**39.2±0.6**	**64.8±0.7**	69.1±0.3
Llama3.1-8B-Inst	Base	10.8±0.2	79.8±0.3	19.2±0.3	51.8±0.3	56.4±0.0
	PPO	9.9±0.1	81.6±0.7	16.1±0.6	57.8±1.3	55.2±1.5
	GRPO	10.0±0.2	80.2±1.1	23.7±0.9	**62.2±2.8**	61.0±1.9
	RandOpt (K=1)	10.7±0.9	80.4±0.6	11.9±0.5	50.8±0.7	55.6±0.8
	RandOpt (K=50)	**63.6±0.4**	**86.7±0.6**	**32.1±0.0**	59.0±0.7	**65.2±0.9**
	CoRP	42.3±0.6	80.6±0.2	20.9±0.5	**60.5±0.4**	61.1±0.8

Models. We use five instruction-tuned models that span 0.5B to 8B parameters and three pretraining lineages: Qwen2.5-0.5B, Qwen2.5-1.5B, and Qwen2.5-3B [Qwen et al., 2025], OLMo3-7B [Olmo et al., 2026], and Llama-3.1-8B [Grattafiori et al., 2024].

Tasks. Each model is evaluated on five tasks. Three of them test mathematical reasoning at increasing difficulty, Countdown [Gandhi et al., 2024], GSM8K [Cobbe et al., 2021], and OlympiadBench [He et al., 2024]. ROCStories tests creative writing [Mostafazadeh et al., 2016], and MBPP tests short Python program synthesis [Austin et al., 2021].

Perturbation population. For each model-task pair we sample N=500 rewarded perturbations using the noise-scale mixture $\Sigma = \{5e{-}4, 1e{-}3, 2e{-}3\}$ from Gan and Isola [2026], and we follow the RandOpt evaluation protocol exactly. Each perturbation is stored by its random seed and noise scale and regenerated on demand, so the population's storage cost is independent of model size. The first 200 training examples are partitioned into the support folds A, B, and P of §4.2, and all results are reported on the original test set.

5.1 Comparison to Standard Post-Training Methods

Table 1 compares CoRP with PPO, GRPO, and two RandOpt references. RandOpt K=1 keeps the best single perturbation, RandOpt K=50 keeps a prediction-level ensemble of the top fifty, and CoRP commits one consolidated update.

Compared with RandOpt K=1, CoRP improves 24 of 25 model-task pairs. It reaches or exceeds RandOpt K=50 in six pairs, including four ROCStories settings, Qwen2.5-0.5B on Countdown, and OLMo3-7B on OlympiadBench. PPO or GRPO gives the best result in a minority of cells.

These results place CoRP between the best single perturbation and the full ensemble. It usually extracts more deployable signal than RandOpt $K{=}1$, while leaving a gap to $K{=}50$ in settings where specialists remain hard to compress.

Table 2 reports the cost at which each method reaches its average score. We measure cost in model-example forward equivalents, omitting the common $2PL$ factor. RandOpt scores $N{=}5000$ perturbations on 200 support examples, for a total of $1{\times}10^6$ forward equivalents. CoRP follows the same sampling procedure with $N{=}500$, while candidate evaluation, validation, probe calibration, and local search bring its complete cost from $1{\times}10^5$ to $1.46{\times}10^5$ forward equivalents, or 14.6% of the RandOpt budget. PPO and GRPO remain matched to RandOpt $K{=}50$ under the same protocol. CoRP improves the base model by 8.1 points, exceeds single-inference RandOpt by 6.5 points, and recovers more than half of the $K{=}50$ ensemble gain. RandOpt $K{=}50$ retains the highest score through fifty inference passes, while CoRP recovers much of that gain with one deployed model and one pass per example.

Table 2: Post-training cost in model-example forward equivalents, inference passes, and average score over 25 model-task pairs. RandOpt uses $N{=}5000$ and CoRP uses $N{=}500$. CoRP cost includes all stages.

Method	Post-train	Inf.	Score
Base	–	1	41.9
PPO	$1{\times}10^6$	1	47.7
GRPO	$1{\times}10^6$	1	48.8
RandOpt $K{=}1$	$1{\times}10^6$	1	43.5
RandOpt $K{=}50$	$1{\times}10^6$	50	**56.3**
CoRP	$\mathbf{1.46{\times}10^5}$	**1**	50.0

5.2 When Can a Rewarded Population Be Consolidated?

A prediction-level ensemble can keep rewarded perturbations separate until inference. A consolidated model cannot. It has to write their useful parts into one set of weights, and any incompatible structure that comes along will be written in too.

We make this tension visible by sweeping two perturbations at a time. For rewarded perturbations Δ_i and Δ_j, we evaluate $\theta_0 + \eta(\alpha\Delta_i + \beta\Delta_j)$ over a grid of mixing coefficients (α, β). The resulting accuracy landscape shows which mixtures survive consolidation.

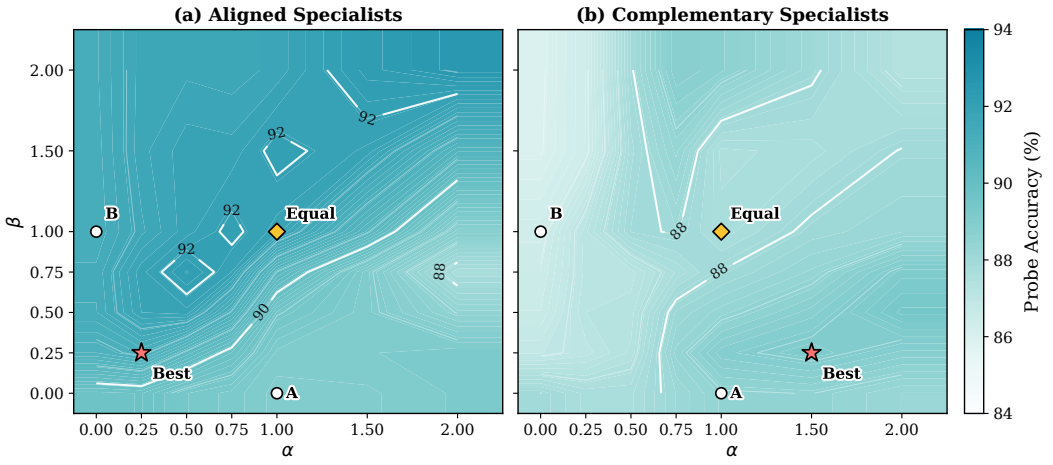


Figure 3: Pairwise consolidation landscapes on Qwen2.5-3B-Instruct and GSM8K. The contours visualize probe accuracy for $\theta_0 + \eta(\alpha\Delta_i + \beta\Delta_j)$. Markers A and B are the individual perturbations, Equal is equal weighting, Best is the best probe point. Panel (a) shows an aligned pair with a broad high-accuracy region. Panel (b) shows a complementary pair with a narrower region and a stronger reweighted point.

Figure 3 shows two characteristic cases on Qwen2.5-3B-Instruct and GSM8K. In panel (a), the two perturbations repair the same probe errors. The high-accuracy region is broad, and equal weighting already lands inside it. This pair behaves like two samples of the same consolidatable component. In panel (b) the repair pattern changes. The two perturbations fix mostly different examples. The high-accuracy region shrinks, and equal weighting falls below the best reweighted point. Complementary perturbations can still be consolidated, but only under more specific weights, and the room for error

is smaller. This is the target of CoRP, to preserve the compatible part of the rewarded population, not every specialist that an ensemble can use separately.

5.3 Composition of the Gain

The same score can hide different behavior. A method may solve examples the base model missed, or it may make an already-correct solution easier for the strict evaluator to parse. It may also improve some examples while regressing on others. Following Gan and Isola [2026], we decompose the test-set outcome into four categories. A *retained correct* example is strictly correct for the base model and remains correct after adaptation. A *format fix* is one where the base produces the right answer content but fails the strict format and the adapted model becomes strictly correct. A *reasoning fix* is one where the base fails on the answer content and the adapted model becomes strictly correct. A *regression* is strictly correct for the base and incorrect after adaptation. Precise definitions and the format-vs-answer distinction per task are in the appendix.

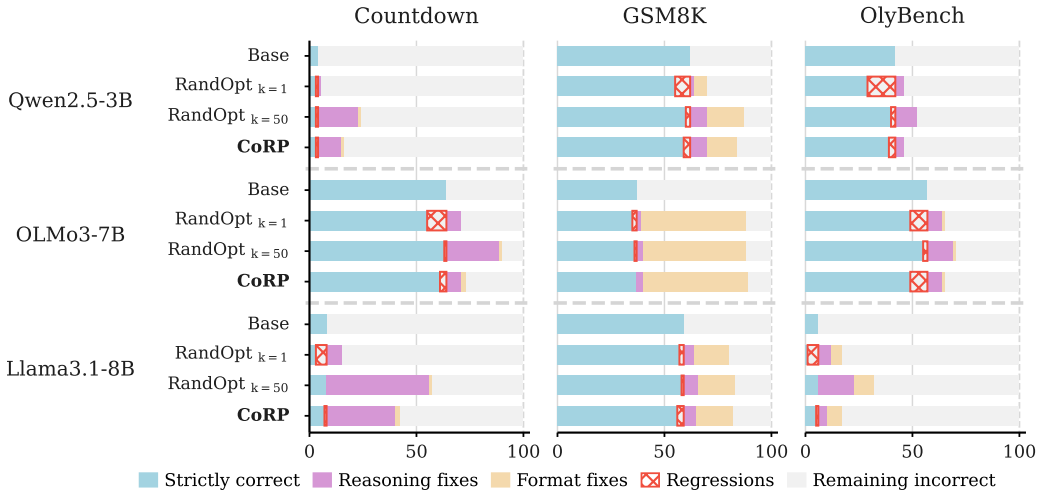


Figure 4: Composition of the test-set outcome on the three math tasks. Each row is one method, and each segment is the fraction of test examples that fall into one of four categories defined in the text. Bars are stacked left to right by strictly correct examples, reasoning fixes, and format fixes. Regressions are overlaid as a hatched negative segment.

Figure 4 shows that CoRP's gain is not only format repair. GSM8K does carry a visible format component for the stronger base models, but Countdown and OlympiadBench show clear reasoning fixes, where the base fails on the answer content and CoRP makes the example strictly correct. This matters because CoRP performs one inference with one updated model, rather than relying on a vote across perturbed models.

The same panels also show why consolidating several rewarded perturbations is different from selecting one. RandOpt $K{=}1$ introduces visible regressions, especially on Countdown and OlympiadBench. CoRP keeps the single-model deployment of $K{=}1$, but its regression segments are smaller in most panels. RandOpt $K{=}50$ can vote regressions away at inference. CoRP cannot. Reducing them before deployment is a central part of consolidation, and the held-out gate of §4.2 is what does it.

Robustness checks. Each component of CoRP carries weight, and full CoRP improves on naive baselines including reward-weighted averaging and direct top-r subspace projection. The choice of compatibility weights γ_a, γ_d is also not fragile in a sweep over $\{0.1, 0.5, 1, 2, 5\}^2$ on Qwen2.5-3B / GSM8K. We report ablations, naive baselines, and the sensitivity sweep in Appendix §A.

6 Related Work

Search in weight space. CoRP is closest in spirit to a line of self-elicitation post-training methods that sample from the model and consolidate the exposure back into the weights, including RLVR and on-policy distillation. Within this lineage, a body of work treats LLM post-training as a search problem

in weight space rather than as gradient descent on a loss. Evolution strategies and the cross-entropy method use reward-weighted samples to update a sampling distribution over weights [Salimans et al., 2017, Mannor et al., 2003, Rubinstein, 1999, Peters and Schaal, 2007, Peng et al., 2019], with covariance-adaptive variants concentrating exploration in informative directions [Hansen, 2023, Maheswaranathan et al., 2019, Choromanski et al., 2019]. RandOpt [Gan and Isola, 2026] pushes this idea to its extreme by removing iterative search entirely. A single round of rewarded Gaussian perturbations with prediction-level ensembling already matches PPO and GRPO on contemporary models. CoRP shares RandOpt's reliance on a single rewarded sampling stage as the primary source of perturbations, but consolidates the rewarded population into one deployable update rather than carrying it forward as an ensemble.

Parameter-space model merging. A second relevant line aggregates several models in parameter space. Model soups average independently fine-tuned checkpoints that share a low-loss basin [Wortsman et al., 2022, Frankle et al., 2020]. When the sources disagree, geometry-aware corrections become necessary. TIES-merging trims small magnitudes and resolves sign conflicts [Yadav et al., 2023], DARE drops and rescales parameters to cut interference [Yu et al., 2024], Fisher-weighted merging weights sources by local curvature [Matena and Raffel, 2022], and task arithmetic operates on task vectors anchored to a shared base [Ilharco et al., 2023]. CoRP applies the same principle to a different source family. Its sources are random rewarded perturbations around one pretrained model rather than independently fine-tuned checkpoints, so compatibility is measured by alignment and dispersion with respect to a provisional reward-weighted mean rather than by sign, magnitude, or curvature.

Low-dimensional structure in LLM adaptation. The stable low-rank signal we measure in rewarded perturbations aligns with a larger body of work on low-dimensional LLM adaptation. Intrinsic-dimension analyses show that fine-tuning succeeds within a small random subspace [Aghajanyan et al., 2021]. LoRA [Hu et al., 2022] and its variants restrict updates to low-rank components without sacrificing accuracy. Liang et al. [2026] attribute the success of LLM adaptation to a low-dimensional curvature geometry, and Morris et al. [2026] show that a mathematical reasoning task can be learned by updating thirteen parameters. Our characterization adds a complementary observation at a different granularity, namely that reward-driven random search around a pretrained model concentrates its useful variance on a consistent low-rank subspace of a coordinate sketch space.

7 Discussion

Post-training can be viewed as an elicitation loop. Output-space methods sample trajectories and use reward or teacher feedback to write useful behavior back into the model [Shao et al., 2024, Agarwal et al., 2024]. Evolution strategies offer a complementary route by using rewards from weight perturbations to update model parameters [Qiu et al., 2026]. RandOpt reveals a related opportunity in weight space, but retains rewarded specialists for inference-time ensembling [Gan and Isola, 2026]. CoRP shows that useful variation across these specialists can instead be consolidated into a single model.

Across all 25 model-task pairs, rewarded perturbations exhibit reproducible low-rank structure even though a stable mean direction is absent in most cases. This suggests that shared geometry need not imply a common update direction. CoRP uses compatibility-aware consolidation to combine rewarded perturbations into a deployable update without any gradient through the language model. It recovers more than half of the 50-pass ensemble gain at 14.6% of RandOpt's post-training cost and one forward pass at inference. More broadly, the result connects weight-space self-elicitation to evidence that language model adaptation often admits low-dimensional structure [Aghajanyan et al., 2021, Liang et al., 2026]. The rewarded neighborhood of a well-pretrained model is therefore not only dense with specialists but also contains enough shared geometry to improve a single model.

Limitations & Future Work. CoRP currently targets settings where a reliable scalar reward is available on a small support set. Our experiments establish consolidation on structured or verifiable tasks. Whether the same population structure can be recovered from preference-based feedback, noisier rewards, or long-horizon outcomes remains open. These settings preserve the underlying consolidation problem, but make it harder to identify which perturbations carry reliable signal.

A second boundary is complementarity. When rewarded perturbations repair different examples, an inference-time ensemble can retain them separately, whereas one checkpoint must reconcile them within a shared parameter region. CoRP captures the compatible component of such populations, but does not preserve every specialist that the ensemble can use. This gap motivates consolidation operators that first represent multiple behavioral modes and then determine which of them can coexist in one update. The validation gate could similarly be extended beyond the target reward to protect unrelated capabilities during consolidation.

Together, these results establish rewarded perturbation populations as a meaningful object for post-training. Their geometry reveals which parts of local weight-space variation are shared, while consolidation tests whether that shared structure can survive in a single model. CoRP provides a first step toward turning this population-level structure into a practical model update.

References

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny

Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, Gang Xie, Hailin Zhang, Hanglong Lv, Hanyu Li, Heyu Chen, Hongshen Xu, Houbin Zhang, Huaqiu Liu, Jiangshan Duo, Jianyu Wei, Jiebao Xiao, Jinhao Dong, Jun Shi, Junhao Hu, Kainan Bao, Kang Zhou, Lei Li, Liang Zhao, Linghao Zhang, Peidian Li, Qianli Chen, Shaohui Liu, Shihua Yu, Shijie Cao, Shimao Chen, Shouqiu Yu, Shuo Liu, Tianling Zhou, Weijiang Su, Weikun Wang, Wenhan Ma, Xiangwei Deng, Bohan Mao, Bowen Ye, Can Cai, Chenghua Wang, Chengxuan Zhu, Chong Ma, Chun Chen, Chunan Li, Dawei Zhu, Deshan Xiao, Dong Zhang, Duo Zhang, Fangyue Liu, Feiyu Yang, Fengyuan Shi, Guoan Wang, Hao Tian, Hao Wu, Heng Qu, Hongfei Yi, Hongxu An, Hongyi Guan, Xing Zhang, Yifan Song, Yihan Yan, Yihao Zhao, Yingchun Lai, Yizhao Gao, Yu Cheng, Yuanyuan Tian, Yudong Wang, Zhen Tang, Zhengju Tang, Zhengtao Wen, Zhichao Song, Zhixian Zheng, Zihan Jiang, Jian Wen, Jiarui Sun, Jiawei Li, Jinlong Xue, Jun Xia, Kai Fang, Menghang Zhu, Nuo Chen, Qian Tu, Qihao Zhang, Qiying Wang, Rang Li, Rui Ma, Shaolei Zhang, Shengfan Wang, Shicheng Li, Shuhao Gu, Shuhuai Ren, Sirui Deng, Tao Guo, Tianyang Lu, Weiji Zhuang, Weikang Zhang, Weimin Xiong, Wenshan Huang, Wenyu Yang, Xin Zhang, Xing Yong, Xu Wang, Xueyang Xie, Yilin Jiang, Yixin Yang, Yongzhe He, Yu Tu, Yuanliang Dong, Yuchen Liu, Yue Ma, Yue Yu, Yuxing Xiang, Zhaojun Huang, Zhenru Lin, Zhipeng Xu, Zhiyang Chen, Zhonghua Deng, Zihan Zhang, and Zihao Yue. Mimo-v2-flash technical report, 2026. URL https://arxiv.org/abs/2601.02780.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=e2TBb5y0yFf.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr, 2026. URL https://arxiv.org/abs/2506.10947.

Jake Ward, Chuqiao Lin, Constantin Venhoff, and Neel Nanda. Reasoning-finetuning repurposes latent representations in base models, 2025. URL https://arxiv.org/abs/2507.12638.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=QGJ9ttXLTy.

Shuozhe Li, Vaishnav Tadiparthi, Kwonjoon Lee, Nakul Agarwal, Hossein Nourkhiz Mahjoub, Ehsan Moradi Pari, Lizhang Chen, Amy Zhang, and Liu Leqi. Learning robust reasoning through guided adversarial self-play. *arXiv preprint arXiv:2602.00173*, 2026a.

Yulu Gan and Phillip Isola. Neural thickets: Diverse task experts are dense around pretrained weights, 2026. URL https://arxiv.org/abs/2603.12228.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/frankle20a.html.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=fq0NaiU8Ex.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=LSKlp_aceOC.

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974.

Greg Lewis and Vasilis Syrgkanis. Double/debiased machine learning for dynamic treatment effects. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=StKuQ0-dltN.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

William Brown. Some experimental results in the correlation of mental abilities 1. *British Journal of Psychology, 1904-1920*, 3(3):296–322, 1910.

Charles Spearman. Correlation calculated from faulty data. *British journal of psychology*, 3(3):271, 1910.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 745–750, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273590. URL https://doi.org/10.1145/1273496.1273590.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Shie Mannor, Reuven Rubinstein, and Yohai Gat. The cross entropy method for fast policy search. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 512–519. AAAI Press, 2003. ISBN 1577351894.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017. URL https://arxiv.org/abs/1703.03864.

Nikolaus Hansen. The cma evolution strategy: A tutorial, 2023. URL https://arxiv.org/abs/1604.00772.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: augmenting random search with surrogate gradients. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4264–4273. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/maheswaranathan19a.html.

Krzysztof M Choromanski, Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, and Vikas Sindhwani. From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/88bade49e98db8790df275fcebb37a13-Paper.pdf.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu,

Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Kanishk Gandhi, Denise H J Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah Goodman. Stream of search (sos): Learning to search in language. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=2cop2jmQVL.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098. URL https://aclanthology.org/N16-1098/.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL https://aclanthology.org/2021.acl-long.568/.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Qiyao Liang, Jinyeop Song, Yizhou Liu, Jeff Gore, Ila Fiete, Risto Miikkulainen, and Xin Qiu. The blessing of dimensionality in llm fine-tuning: A variance-curvature perspective, 2026. URL https://arxiv.org/abs/2602.00170.

John X. Morris, Niloofar Mireshghallah, Mark Ibrahim, and Saeed Mahloujifar. Learning to reason in 13 parameters, 2026. URL https://arxiv.org/abs/2602.04118.

Xin Qiu, Yulu Gan, Conor F. Hayes, Qiyao Liang, Yinggan Xu, Roberto Dailey, Elliot Meyerson, Babak Hodjat, and Risto Miikkulainen. Evolution strategies at scale: LLM fine-tuning beyond reinforcement learning. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 101987–102015. PMLR, 06–11 Jul 2026. URL https://proceedings.mlr.press/v306/qiu26f.html.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Yuanzhe Liu, Jingyuan Zhu, Yuchen Mo, Gen Li, Xu Cao, Jin Jin, Yifan Shen, Zhengyuan Li, Tianjiao Yu, Wenzhen Yuan, et al. Palm: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. *arXiv preprint arXiv:2601.07060*, 2026.

Boyi Li, Yifan Shen, Yuanzhe Liu, Yifan Xu, Jiateng Liu, Xinzhuo Li, Zhengyuan Li, Jingyuan Zhu, Yunhan Zhong, Fangzhou Lan, et al. Toward cognitive supersensing in multimodal large language model. *arXiv preprint arXiv:2602.01541*, 2026b.

Yifan Shen, Yuanzhe Liu, Jingyuan Zhu, Xu Cao, Xiaofeng Zhang, Yixiao He, Wenming Ye, James Rehg, and Ismini Lourentzou. Fine-grained preference optimization improves spatial reasoning in vlms. *Advances in Neural Information Processing Systems*, 38:17929–17960, 2026.

Tianjiao Yu, Xinzhuo Li, Yifan Shen, Yuanzhe Liu, and Ismini Lourentzou. Core3d: Collaborative reasoning as a foundation for 3d intelligence. *arXiv preprint arXiv:2512.12768*, 2025.

Yifu Luo, Zeyu Chen, Haoyu Wang, Xinhao Hu, Yuxuan Zhang, Zhizhou Sha, and Shiwei Liu. Learning from the self-future: On-policy self-distillation for dllms. *arXiv preprint arXiv:2606.18195*, 2026a.

Yifu Luo, Xinhao Hu, Keyu Fan, Haoyuan Sun, Zeyu Chen, Bo Xia, Tiantian Zhang, Yongzhe Chang, and Xueqian Wang. Reinforcement learning meets masked generative models: Mask-grpo for text-to-image generation. *Advances in Neural Information Processing Systems*, 38:108460–108485, 2026b.

Zhen-Yu Zhang, Jiandong Zhang, Huaxiu Yao, Gang Niu, and Masashi Sugiyama. Dataset mismatch matters in group relative policy optimization for reinforcement learning from verifiable rewards, 2026. URL https://arxiv.org/abs/2611.00586.

ModelScope Team. EvalScope: Evaluation framework for large models, 2024. URL https://github.com/modelscope/evalscope.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

A Additional Empirical Results

This section reports four additional empirical studies that complement the main results: a component ablation of CoRP on representative cells, a comparison to naive and merging baselines on the same cells, a sensitivity sweep of the compatibility weights, and an analysis of the effect of perturbation population size.

A.1 Component Ablation

We isolate the contribution of each CoRP component on four cells covering two model scales and two task types. Five variants share the rewarded population and the support folds with full CoRP, and differ only in which component is removed. The variants are reward only (γ_a=γ_d=0, a single reward-weighted average), reward+alignment (γ_d=0), reward+dispersion (γ_a=0), no gate (deploys the best-on-A candidate without the validation of §4.2), and no iteration (skips the recentering of §4.3). Table 3 reports test accuracy and the change relative to the base model.

Table 3: Component ablation on Qwen2.5-3B-Instruct and OLMo3-7B-Instruct, across GSM8K and ROCStories. Each entry reports test accuracy and, in parentheses, the change relative to the base model on the same cell. Full CoRP corresponds to the configuration of Table 1.

	GSM8K		ROCStories	
Method	Acc.	Δ	Acc.	Δ
Qwen2.5-3B-Instruct				
Base	79.81	–	54.73	–
reward only	80.97	+1.16	54.70	−0.03
reward+alignment	80.89	+1.08	53.97	−0.76
reward+dispersion	79.76	−0.05	54.66	−0.07
no gate	80.67	+0.86	54.59	−0.14
no iteration	**82.31**	**+2.50**	56.71	+1.98
Full CoRP	**82.31**	**+2.50**	**59.11**	**+4.38**
OLMo3-7B-Instruct				
Base	82.92	–	64.04	–
reward only	84.16	+1.24	63.31	−0.73
reward+alignment	84.69	+1.77	62.82	−1.22
reward+dispersion	84.57	+1.65	63.40	−0.64
no gate	**85.63**	**+2.71**	63.66	−0.38
no iteration	84.82	+1.90	64.61	+0.57
Full CoRP	84.82	+1.90	**64.81**	**+0.77**

Each component contributes. The iteration step helps most on Qwen2.5-3B / ROCStories, where Full CoRP improves over the no-iteration variant by 2.40 points. On the other three cells, the accepted first-pass update already captures most of the measured gain. The held-out gate plays a complementary role by rejecting harmful candidates on cells where rewarded perturbations are less aligned. Dispersion is the weakest stand-alone choice, which matches its role in Eq. 9: it suppresses incompatible mass against a direction the alignment term has already proposed, and is informative only when paired with that direction.

A.2 Naive Baselines

The reproducible low-rank structure of §3 raises a natural concern: simpler operators that read this structure directly might already suffice. Existing parameter-space merging tools, designed for fine-tuned checkpoints, raise a separate concern about whether they transfer to rewarded perturbations. We compare CoRP to three alternatives that probe both concerns on the same four cells used in §A.1.

Reward-weighted averaging is the operator obtained by setting γ_a=γ_d=0 in Eq. 9, identical to the reward-only variant of Table 3. *Top-r subspace projection* estimates the top-r principal subspace of the elite perturbations with r=8 and deploys $\theta_0 + \eta\,\Pi_U\bar{\Delta}$, where Π_U projects onto the estimated subspace. *Sparse merging in the spirit of TIES* [Yadav et al., 2023] selects the top half of the rewarded population, sparsifies each perturbation by retaining the top-density fraction of coordinates by magnitude, and merges the sparsified vectors. We sweep the sparsity density and deployment scale

on the probe fold for each cell. All three baselines share the rewarded population and the support folds with full CoRP. Table 4 reports the result.

Table 4: Naive and merging baselines on Qwen2.5-3B-Instruct and OLMo3-7B-Instruct, across GSM8K and ROCStories. Each entry reports test accuracy and, in parentheses, the change relative to the base model on the same cell.

	GSM8K		ROCStories	
Method	Acc.	Δ	Acc.	Δ
Qwen2.5-3B-Instruct				
Base	79.81	–	54.73	–
reward-weighted avg.	80.97	+1.16	54.70	−0.03
top-r projection	80.59	+0.78	54.59	−0.14
sparse merging (TIES-style)	80.42	+0.61	53.91	−0.82
Full CoRP	**82.31**	**+2.50**	**59.11**	**+4.38**
OLMo3-7B-Instruct				
Base	82.92	–	64.04	–
reward-weighted avg.	84.16	+1.24	63.31	−0.73
top-r projection	83.85	+0.93	63.85	−0.19
sparse merging (TIES-style)	83.32	+0.40	62.32	−1.72
Full CoRP	**84.82**	**+1.90**	**64.81**	**+0.77**

Reward-weighted averaging is the most competitive of the three baselines, but still trails CoRP across the four cells. Reward identifies useful perturbations, but compatibility against the proposal determines which of those perturbations can share a single deployable update.

Top-r projection rarely moves beyond the base. The shared low-rank structure of §3 is a necessary condition for consolidation to be possible, but the population-level subspace estimate discards the per-perturbation reward signal that the operator of §4.1 actually uses, and the estimate is itself noisy at the population sizes we use.

Sparse merging in the spirit of TIES underperforms the base on two of four cells. Standard merging operators were designed for independently fine-tuned checkpoints whose disagreements carry meaningful sign and magnitude structure. The same operators applied to random rewarded perturbations have no comparable signal to read, since coordinate sign is essentially random and magnitude varies with the noise scale rather than with task relevance.

A.3 Sensitivity to Compatibility Weights

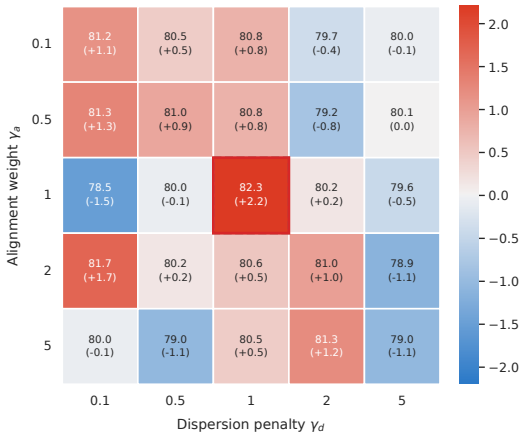


Figure 5: Test accuracy on Qwen2.5-3B-Instruct / GSM8K under the first-pass CoRP operator, swept over the alignment weight γ_a and dispersion penalty γ_d. Each cell reports accuracy and, in parentheses, the change relative to the base model (base 80.06). The default $\gamma_a{=}\gamma_d{=}1$ is the best configuration on this sweep.

The default configuration of CoRP fixes the alignment and dispersion weights at $\gamma_a{=}\gamma_d{=}1$ across all 25 model-task pairs. To check that this choice is not fragile, we sweep both weights over $\{0.1, 0.5, 1, 2, 5\}$ on Qwen2.5-3B-Instruct / GSM8K, holding the rewarded population, the support folds, and the candidate grid over (q, β, α) identical to the main run. Each of the 25 configurations rebuilds the consolidated update of §4.1 from the same perturbations and is evaluated on the GSM8K test set. The sweep uses the first-pass operator only, without the held-out gate of §4.2 or the iteration of §4.3, so that the variation in the table reflects only the choice of (γ_a, γ_d).

The default $(1, 1)$ is the strongest configuration on the sweep, and the central region $[0.5, 2]^2$ remains mostly positive, with small negative outliers at a few asymmetric settings. The accuracy surface drops at the corners where one weight saturates the exponent in Eq. 9 and the other becomes negligible, which is the regime in which compatibility scoring effectively reduces to one of the two terms. Within the operating range we use, the operator is not sensitive to the precise value of either weight.

A.4 Effect of Perturbation Population Size

CoRP uses $N{=}500$ rewarded perturbations throughout the main results, one tenth of the 5000 used by RandOpt. To check that this choice is not arbitrary, we sweep $N \in \{100, 250, 500, 1000, 5000\}$ on Qwen2.5-3B-Instruct, holding the noise-scale mixture, the support folds, and the candidate grid identical to the main run. Each configuration runs the first-pass operator of §4.1, without the iteration of §4.3, so that the variation in Figure 6 reflects only the population size.

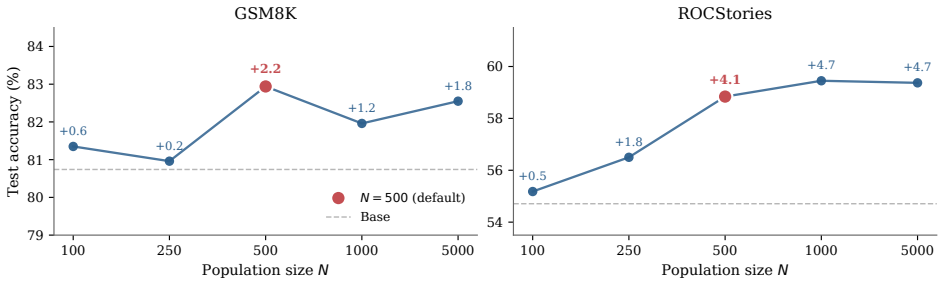


Figure 6: CoRP test accuracy on Qwen2.5-3B-Instruct as a function of perturbation population size N. The dashed line marks the base model and the parenthesized number annotates the gain over base. The default $N{=}500$ used in the main results is highlighted in red. The sweep uses the first-pass operator only.

The marginal return from a larger population diminishes well before $N{=}5000$ on both tasks. On ROCStories, the gain plateaus between $N{=}1000$ and $N{=}5000$. On GSM8K, the gain peaks at $N = 500$ in this sweep and remains positive but non-monotone for larger populations. The default $N{=}500$ recovers a substantial fraction of the maximum gain on both tasks at one tenth of the population size used by RandOpt.

B Additional Analysis on Consolidation Behavior

We provide two further analyses of consolidation behavior: how CoRP composes its test-set outcomes relative to ensemble baselines, and how strict vs. relaxed evaluation interacts with each method.

B.1 Error-Composition Decomposition

Figure 7 reveals a structured trade-off across the three deployment modes. Relative to $K{=}1$, CoRP reduces the average regression fraction while preserving a comparable reasoning-thicket fraction, indicating that compatibility-aware reweighting and held-out gating mitigate harmful updates without sacrificing reasoning gains. The 50-pass vote attains the lowest regression level by retaining inference-time diversity, but at substantially higher inference cost. CoRP therefore lands between the two on the regression-vs-inference-cost frontier, achieving the regression reduction of an ensemble at the inference cost of a single model.

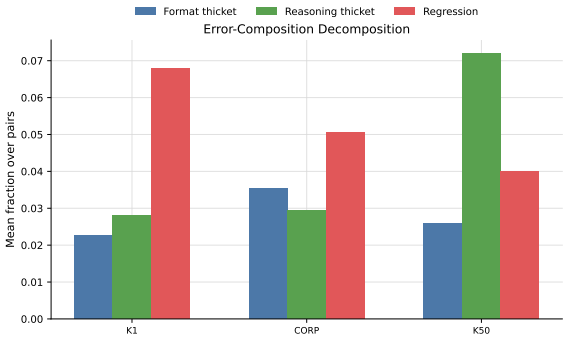


Figure 7: Mean bucket fractions across model-task pairs for RandOpt single-inference ($K{=}1$), CoRP, and 50-pass majority vote ($K{=}50$).

B.2 Robustness to Strict Evaluation

Figure 8 reports how each method's gain holds up under the strict task evaluator used throughout the paper. The 50-pass vote has the most negative average strict-minus-relaxed gap, suggesting that part of its reward-level gain is attenuated under strict parsing. CoRP remains closer to the single-inference baseline on this axis, indicating that consolidation does not systematically amplify strict-format fragility. For deployment with strict extractors this distinction is practically important, because the gap between an ensemble's relaxed score and its deployed strict score is exactly what a single deployable model has to recover.

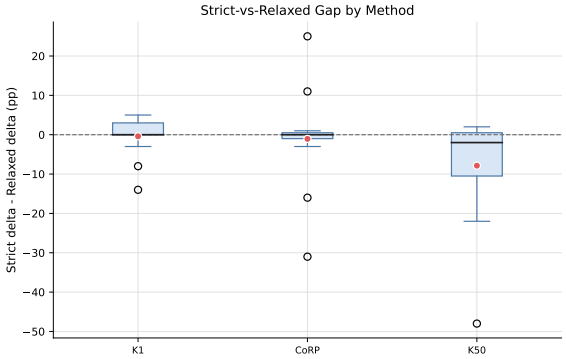


Figure 8: Distribution of strict-minus-relaxed deltas (percentage points) across model-task pairs. More negative values indicate that a method's gain shrinks more under strict parsing relative to relaxed evaluation.

C Implementation Details

We implement CoRP in PyTorch [Paszke et al., 2019] and all experiments were conducted on 4 NVIDIA A100 GPUs with 80GB memory.

C.1 Support Splits and Model Selection

For each model-task pair, we partition the support set into three disjoint folds: a construction fold A, a validation fold B, and a probe fold P. Fold A scores perturbations and generates CoRP candidates. Fold B gates candidates and selects among the (q, β) grid. Fold P is used only after a candidate passes the gate, to choose the final step-size multiplier α from the fixed grid in Table 5. The test set plays no role in selecting q, β, or α, nor in deciding whether to accept an update.

This design separates proposal construction from validation. If no candidate achieves a positive construction score and a positive lower-confidence-bound improvement on fold B (defined in §C.5), CoRP abstains and returns the base model. If a candidate passes the gate but no step-size multiplier passes the probe check on P, CoRP also abstains.

C.2 Baseline Protocols

Post-training helps adapt LLMs to a broad range of tasks and domains [Ouyang et al., 2022, Shao et al., 2024, Agarwal et al., 2024, Liu et al., 2026, Li et al., 2026b, Shen et al., 2026, Yu et al., 2025, Luo et al., 2026a,b, Zhang et al., 2026]. We compare CoRP against PPO, GRPO, RandOpt $(K = 1)$, and RandOpt $(K = 50)$ under the protocol of Gan and Isola [2026]. RandOpt $(K = 1)$ selects the single highest-reward perturbation from the sampled population; RandOpt $(K = 50)$ ensembles the top 50 perturbations by majority vote at inference. RandOpt uses $N = 5000$ perturbations and CoRP uses $N = 500$. For PPO and GRPO, we follow the matched-compute settings reported by Gan and Isola [2026]: total training FLOPs are matched to those consumed by sampling and scoring the RandOpt K=50 population, accounting for actor forward and backward passes, KL evaluation against the reference model, and rollout sequence lengths. We use the same models, tasks, prompts, reward functions, and evaluation scripts throughout. No baseline hyperparameters are tuned on the test set.

C.3 Statistical Reporting

The main result table reports mean±standard deviation over $R = 3$ independent runs. For CoRP and RandOpt, each run resamples the perturbation population and repeats the full selection procedure; for PPO and GRPO, each run uses an independent training seed under the same matched-compute setting. Error bars summarize run-to-run variability and are not intended as formal pairwise hypothesis tests.

For the split-half diagnostics of §3, we repeat the random partition of the top-M rewarded perturbations 20 times and report a nonparametric 95% lower confidence bound via bootstrapping, taking the 5th percentile of the bootstrap distribution of the mean. This procedure applies to both the mean-consensus statistic and the subspace-excess statistic, and is distinct from the gate-and-probe lower confidence bounds defined in §C.5.

C.4 Hyperparameter Settings

Table 5 lists the CoRP hyperparameters. Values are fixed across all 25 model-task pairs unless noted.

Table 5: The hyperparameter configuration of our experiments. Values are fixed across model-task pairs unless noted.

Symbol	Value	Notes
(A) Sampling		
N	500	rewarded perturbations per pair
Σ	$\{5\times10^{-4},\ 1\times10^{-3},\ 2\times10^{-3}\}$	noise-scale mixture, uniform per perturbation
$\lvert A\rvert, \lvert B\rvert, \lvert P\rvert$	$75, 75, 50$	first 200 training examples per task
(B) Candidate and gate		
q	$\{0.5,\ 0.7,\ 0.9\}$	elite quantile, selected on B
β	$\{0.5,\ 1,\ 2,\ 5,\ 10,\ 20,\ 50\}$	inverse temperature, selected on B
$\gamma_a,\ \gamma_d$	$1,\ 1$	alignment and dispersion weights
λ	2.0	regression penalty in the constructive score
z_{LCB}	1.645	one-sided 95% normal-approximation LCB
$\{\alpha_j\}$	$\{0.5,\ 1,\ 2,\ 4,\ 8,\ 16\}$	probe multiplier grid, selected on P
(C) Iteration		
N_{loc}	100	local perturbations per iteration
r	8	truncation rank of Σ_t
λ_{iso}	0.5	isotropic floor, clipped to $[0.05, 0.95]$

C.5 Operational Definitions

We give precise definitions for three quantities referenced in the main text: the coordinate sketch used by the split-half diagnostic, the proposal covariance Σ_t used by the iteration step, and the lower confidence bounds used by the validation gate and probe.

Coordinate sketch. The split-half diagnostic of §3 reads each statistic from a fixed low-dimensional sketch rather than from $\mathbb{R}^d$ directly. For each perturbable parameter tensor in the model, we retain a small fixed number of coordinates from the flattened perturbation, and concatenate the retained coordinates across all tensors to form $z_i \in \mathbb{R}^m$ with m on the order of a few thousand. The same coordinate selection is shared across all candidates in the population and across all 20 random splits. We use the same sketch when comparing reward-weighted means and when computing principal subspaces, normalizing z_i by its noise scale σ_i to compare directions rather than magnitudes.

Proposal covariance. The iteration step of §4.3 draws local perturbations $\Delta_j^{\mathrm{loc}} \sim \mathcal{N}(0, \rho_t^2 \Sigma_t)$, where the covariance Σ_t combines a low-rank component fit to the previous accepted population with an isotropic floor:

$$\Sigma_t \;=\; \lambda_{\mathrm{iso}}\, I_d \;+\; (1 - \lambda_{\mathrm{iso}})\, U_t \Lambda_t U_t^{\top}, \tag{13}$$

where (U_t, Λ_t) is the rank-r truncated eigendecomposition of the compatibility-weighted second-moment matrix of the previous round's elite perturbations, and the rank r and floor λ_{iso} are reported in Table 5. The covariance shapes the local proposal distribution only and does not appear in the deployed update.

Lower confidence bounds. The gate and probe of §4.2 commit a candidate only when its estimated accuracy change has a positive lower confidence bound on the corresponding fold. For each candidate, we compute paired per-example correctness differences against the base model and use

$$\widehat{\mathrm{LCB}} = \widehat{\Delta\mathrm{Acc}} - z_{\mathrm{LCB}}\, \widehat{\mathrm{SE}}(\Delta\mathrm{Acc}). \tag{14}$$

where z_{LCB} is given in Table 5. This bound is computed separately for each candidate and fold, and is distinct from the bootstrap procedure used for the split-half diagnostics in §3.

C.6 Prompts

Following Gan and Isola [2026], we set up the prompts for different datasets in our experiments following EvalScope [Team, 2024] and Verl [Sheng et al., 2024].

Countdown

Your Task
Using the numbers `{numbers}`, create an equation that equals `{target}`.

Instructions
You can use basic arithmetic operations (`+`, `-`, `*`, `/`), and each number can only be used once.
Show your work in `<work>` ... `</work>` tags.
Return the final answer in `<answer>` ... `</answer>` tags.
For example: `<answer> (1 + 2) / 3 </answer>`.

GSM8K / OlympiadBench

Your Task
Solve the following problem: `{question}`.

Instructions
Let's think step by step and output the final answer after `####`.

ROCStories

Your Task
Below are 5 sentences from a story, but they are in the wrong order.
Please arrange them in the correct chronological order.

Story
Title: `{title}`
Sentence A: `{sentence1}`
Sentence B: `{sentence2}`
Sentence C: `{sentence3}`
Sentence D: `{sentence4}`
Sentence E: `{sentence5}`

Instructions
Output the correct order as comma-separated letters (e.g., `B,A,D,E,C`).
Only output the letters, nothing else.

MBPP

Your Task
You are an expert Python programmer, and here is your task: `{question}`

Instructions
Your code should pass these tests: `{tests}`.

D Additional Acknowledgment

The authors acknowledge the use of ChatGPT exclusively to refine the text in the final manuscript.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in §7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the information needed to reproduce the main experimental results in the Appendix §C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the data and code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the experimental details in the Appendix §C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard deviation in our experiment results and Appendix §C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were conducted on 4 NVIDIA A100 GPUs with 80GB memory (mentioned in Appendix §C).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of our method in §7.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not pose such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the datasets used in our paper are cited accordingly.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not provide any assets in our work.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We have clearly stated the usage of LLMs in our work

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.