CONSENSUS-GUIDED PREFERENCE OPTIMIZATION FOR MULTI-SOURCE LANGUAGE MODEL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models often possess complementary capabilities, yet combining those capabilities in one deployable model remains difficult when source models differ in architecture, specialization, and probability calibration. Most fusion approaches rely on parameter merging or supervised fine-tuning. Existing preference-stage alternatives depend critically on the quality of constructed chosen–rejected pairs. We introduce Consensus-Guided Preference Optimization (CGPO), a framework for multi-source language model alignment from unpaired preference supervision. CGPO turns source-model sequence probabilities into a confidence-calibrated consensus that directly guides each pointwise preference update. The consensus supplies a completion-specific optimization baseline, while pointwise verifier feedback identifies desirable and undesirable responses. This design transfers complementary capabilities from heterogeneous sources without parameter merging, vocabulary alignment, or artificially constructed preference pairs. Comprehensive experiments on seven benchmarks spanning mathematics, code generation, reasoning, and instruction following show that CGPO outperforms existing model-fusion and preference-optimization baselines. For Phi-3.5, CGPO raises the average performance across these benchmarks from 58.1 to 63.3. Our code is available at `https://anonymous.4open.science/r/CGPO-6501/`.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated broad capabilities, while a growing range of domain-specialized models now excels in areas such as mathematical reasoning, code generation, and instruction following (Chen et al., 2021; Ouyang et al., 2022; Yang et al., 2024). These expert models offer complementary strengths, but deploying and invoking multiple experts together incurs substantial memory, computation, and inference overhead. Model fusion addresses this tension by transferring complementary knowledge from multiple source models into a single deployable target (Wan et al., 2024a; Yang et al., 2025; Gu et al., 2025). By retaining one target at inference time, it seeks broader capability coverage without the cost and coordination burden of simultaneously deploying every source.

Existing methods realize this knowledge transfer at different training stages. *Parameter-space merging* combines model weights or task vectors into one checkpoint without adding inference-time models, but generally assumes compatible architectures and parameter spaces (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023). *SFT-stage fusion* trains the target on source responses or predictive distributions. It supports heterogeneous sources, but transfers their knowledge through supervised imitation and may require token-level alignment when distilling distributions (Wan et al., 2024a;b). *Preference-stage fusion* instead incorporates source responses or sequence-level probabilities directly into preference alignment (Yang et al., 2025; Gu et al., 2025). Operating on responses or sequence-level scores allows it to accommodate sources with different architectures and parameterizations, avoiding the compatibility requirements of parameter-space merging. Moreover, source-derived quality distinctions enter the alignment objective directly, providing richer supervision than transferring source behavior through SFT imitation alone. Preference-stage fusion thus combines architectural flexibility with preference-aware transfer, motivating our focus on this setting.

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

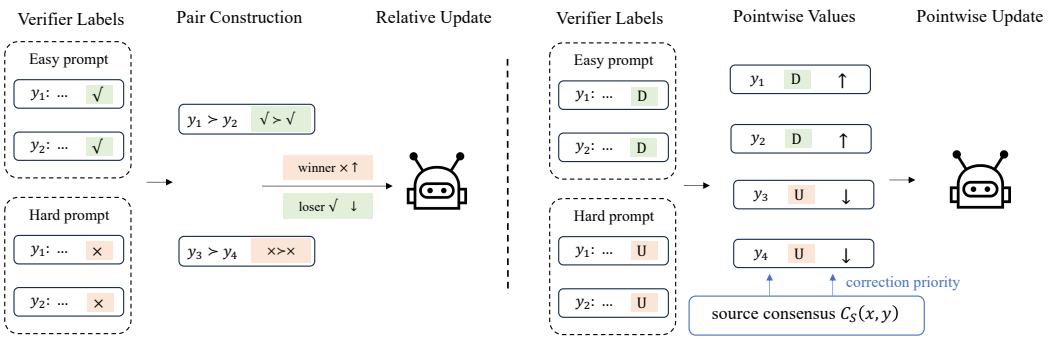


Figure 1: Comparison between existing pairwise preference fusion and CGPO. (a) Pairwise methods impose relative winner–loser updates even when both completions share the same verifier label. (b) CGPO preserves pointwise desirable/undesirable supervision and uses source consensus to guide the correction priority of undesirable completions.

Despite these advantages, existing preference-stage fusion methods generally formulate source transfer through paired chosen–rejected comparisons. Pairwise supervision encodes a relative ordering, but candidate pools across task domains may contain responses of comparable quality. Assigning fixed winner and loser roles can then obscure the supervision associated with each completion. This limitation raises a broader question: *How can we systematically fuse knowledge from multiple source models under unpaired preference supervision?*

KTO (Ethayarajh et al., 2024) offers a natural starting point because it directly optimizes independently labeled desirable and undesirable completions. Yet adapting KTO to multi-source fusion raises two key challenges. First, *how should heterogeneous source models participate in a pointwise value objective originally built around a single reference model?* Second, *how can their probabilistic signals be made reliable despite differences in expertise, confidence, and score offsets?*

To address these challenges, we propose CGPO, a framework for **C**onsensus-**G**uided **P**reference **O**ptimization. CGPO retains KTO's desirable/undesirable supervision and computes source consensus for each completion. The consensus neither replaces the verifier's correctness judgment nor defines a distribution to imitate; after calibration, it serves as a completion-specific source reference. Verifier labels determine the update direction, the initial policy anchors desirable completions, and the source reference determines the correction priority of undesirable ones. CGPO therefore transfers complementary source capabilities without constructing artificial chosen–rejected pairs. Figure 1 contrasts conventional pairwise preference-stage fusion with CGPO's consensus-guided pointwise formulation.

CGPO has two main components. **Source-Consensus Value Guidance** addresses the first challenge by incorporating multi-source probability evidence into the value objective: verifier-labeled D/U completions determine the learning direction, while source consensus modulates the correction priority of undesirable completions. **Consensus Calibration and Stabilization** addresses the second challenge by making this baseline reliable: it uses entropy-based confidence weighting to calibrate expert contributions, prompt-level source–target baseline alignment to remove additive score offsets, and D-first/U-warmup to stabilize early undesirable-loss optimization. Together, these components turn source consensus from a naïve aggregation of expert predictions into a confidence-aware, offset-calibrated, and optimization-stable training signal.

In summary, we introduce a pointwise perspective on preference-stage model fusion: verifier labels specify whether each completion should be promoted or suppressed, while multi-source consensus provides completion-specific guidance without constructing response pairs. We instantiate this idea in **CGPO** through the two components described above. Source-Consensus Value Guidance separates label-defined learning direction from source-guided correction priority, whereas Consensus Calibration and Stabilization makes this guidance reliable through confidence-calibrated aggregation, prompt-level source–target baseline alignment, and D-first/U-warmup. Across seven bench-

marks spanning mathematics, code, reasoning, and instruction following, CGPO achieves the best overall average among fusion and preference-optimization baselines, improving Phi-3.5 from 58.1 to 63.3.

2 Related Work

2.1 Model Fusion

Model fusion consolidates complementary capabilities into one deployable model. Parameter-space methods combine checkpoints or task vectors through weighted averaging, Fisher weighting, task arithmetic, or sign-conflict resolution (Wortsman et al., 2022; Matena & Raffel, 2022; Ilharco et al., 2023; Yadav et al., 2023). They are inference-efficient but typically require compatible parameter spaces. LLM-Blender instead ranks and fuses responses at inference time, retaining multi-model deployment (Jiang et al., 2023). Distillation trains a target to imitate teacher outputs or distributions; FuseLLM and FuseChat extend this strategy to heterogeneous LLMs (Hinton et al., 2015; Wan et al., 2024a;b). CGPO instead transfers sequence-level source consensus during preference alignment, without weight or token-level distribution merging.

2.2 Pairwise and Pointwise Preference Optimization

RLHF learns a reward from comparisons and optimizes a reference-regularized policy (Christiano et al., 2017; Ouyang et al., 2022). Direct objectives include DPO's Bradley–Terry loss over chosen–rejected policy–reference log-ratios (Bradley & Terry, 1952; Rafailov et al., 2023) and RRHF's response-ranking loss (Yuan et al., 2023); RSO instead rejection-samples pairs toward an estimated optimal policy (Liu et al., 2024). IPO, ORPO, and SimPO modify the margin, likelihood, or normalization used in pairwise optimization (Gheshlaghi Azar et al., 2024; Hong et al., 2024; Meng et al., 2024). Pair construction, however, discards absolute verifier judgments and need not yield a genuine desirable–undesirable contrast on very easy or difficult prompts.

KTO and BCO instead learn from independently labeled desirable and undesirable completions using prospect-theoretic values and binary classification, respectively (Kahneman & Tversky, 1979; Ethayarajh et al., 2024; Jung et al., 2025). KTO treats the policy–reference log-ratio as an outcome and evaluates its gain or loss relative to a KL-derived reference point, thereby promoting desirable completions and suppressing undesirable ones without pairs. CGPO retains this pointwise supervision while introducing multi-source sequence probabilities into the value assessment.

2.3 Preference-Stage Model Fusion

WRPO constructs hybrid tuples of source-preferred, target-preferred, and target-rejected responses, then progressively shifts reward weight toward the source preference to bridge their distribution gap (Yang et al., 2025). InfiFPO scores chosen and rejected responses under each source, replaces DPO's reference with their fused sequence-level distribution, and uses normalization, clipping, and max-margin fusion for robustness (Gu et al., 2025). Both transfer source capabilities through pairs; CGPO instead guides a pointwise value objective with source consensus.

3 Preliminaries

3.1 Problem Formulation

We study model fusion under unpaired preference supervision. Let π_0 be the initial target policy (obtained by supervised fine-tuning in our experiments) and $\mathcal{S} = \{q_i\}_{i=1}^{N}$ source models that may differ in scale, tokenizer, architecture, and specialization. We seek an improved, independently deployable policy π_θ. Sources participate only in data construction and offline scoring and are not required at inference time.

The training data consists of independently labeled completions

$$\mathcal{D} = \{(x_j, y_j, z_j)\}_{j=1}^{M}, \qquad z_j \in \{D, U\},$$

where x_j is a prompt, y_j a completion, and z_j an independently assigned desirable (D) or undesirable (U) label from a judge, verifier, reward model, or executable checker. Because labels attach to individual completions, responses sharing a prompt need not form chosen–rejected pairs.

For any policy p, we write the completion-only, length-normalized log probability as

$$\ell_p(x, y) = \frac{1}{|y|} \sum_{t=1}^{|y|} \log p(y_t \mid x, y_{<t}),$$

where the average includes only completion tokens. Each source scores the same text with its own tokenizer, making the sequence-level scores comparable without token-level vocabulary alignment. The learning problem is to use $\{\ell_{q_i}(x, y)\}_{i=1}^{N}$ to guide the update from π_0 to π_θ while retaining pointwise supervision.

3.2 Kahneman–Tversky Optimization

Kahneman–Tversky Optimization (KTO) learns from independently labeled completions (Ethayarajh et al., 2024). Its prospect-theoretic premise is that an outcome is perceived as a gain or loss relative to a reference point (Kahneman & Tversky, 1979). KTO treats the completion-level policy–reference log-ratio as the outcome and the policy's overall KL departure as that reference point:

$$r_\theta(x, y) = \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}, \qquad z_0 = \mathrm{D}_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)).$$

In practice, z_0 is estimated by a detached shared minibatch statistic. For $\beta > 0$ and label weights $\lambda_D, \lambda_U > 0$, KTO uses

$$v_\theta(x, y, z) = \begin{cases} \lambda_D \, \sigma(\beta(r_\theta(x, y) - z_0)), & z = D, \\ \lambda_U \, \sigma(\beta(z_0 - r_\theta(x, y))), & z = U, \end{cases} \tag{1}$$

$$\mathcal{L}_{\mathrm{KTO}}(\theta) = \mathbb{E}_{(x,y,z)\sim\mathcal{D}} \left[\lambda_z - v_\theta(x, y, z)\right], \qquad \lambda_z = \begin{cases} \lambda_D, & z = D, \\ \lambda_U, & z = U. \end{cases} \tag{2}$$

Here $r_\theta - z_0$ asks whether support for a desirable completion increases beyond the policy's typical drift, whereas $z_0 - r_\theta$ rewards reducing support for an undesirable one. Labels therefore determine the update direction, while the shared reference point prevents indiscriminate policy movement. The sigmoid provides diminishing sensitivity at extreme margins, β controls its saturation, and λ_D, λ_U balance the two labels. CGPO retains this pointwise structure but augments its single-policy reference with multi-source evidence.

4 The Proposed Method: CGPO

Consensus-Guided Preference Optimization (CGPO) separates learning direction from correction priority: pointwise verifier labels determine whether to promote or suppress a completion, while source consensus prioritizes undesirable corrections. Figure 2(a) shows source and target completions receiving independent labels without pair construction. Figure 2(b) shows the label-conditional guidance: the frozen initialization anchors desirable completions, while source consensus determines how strongly to correct undesirable ones. Figure 2(c) summarizes the calibration and curriculum that make this guidance reliable during training.

4.1 Source-Consensus Value Guidance

A single reference cannot distinguish a verified error that experts jointly reject from one they still support. For source models $\mathcal{S} = \{q_i\}_{i=1}^{N}$, we therefore define

$$s_i(x, y) = \ell_{q_i}(x, y), \tag{3}$$

$$C_{\mathcal{S}}(x, y) = \sum_{i=1}^{N} w_i(x, y) s_i(x, y), \qquad w_i(x, y) \geq 0, \quad \sum_{i=1}^{N} w_i(x, y) = 1. \tag{4}$$

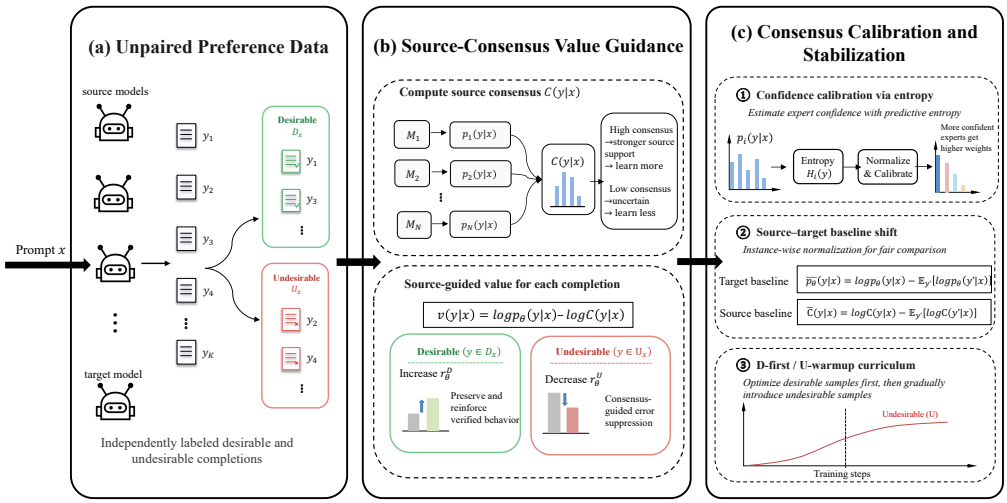


Figure 2: Overview of CGPO. (a) Source and target models generate completions that are independently labeled as desirable or undesirable, avoiding pair construction. (b) The frozen initialization anchors desirable completions, whereas source-model sequence scores form a consensus that guides the suppression of undesirable completions. (c) Entropy-based confidence calibration, prompt-level source–target baseline alignment, and the D-first/U-warmup curriculum improve the reliability of consensus guidance and stabilize training.

The weights are calibrated in Section 4.2; $C_{\mathcal{S}}(x, y)$ measures expert support for a completion, rather than its correctness. Since raw source and target scores may have different prompt-dependent offsets, Section 4.2 aligns $C_{\mathcal{S}}$ to the initial policy and yields the source reference $\widetilde{C}_{\mathcal{S}}$.

CGPO uses label-conditional implicit rewards

$$r_\theta^D(x, y) = \ell_\theta(x, y) - \ell_{\pi_0}(x, y), \tag{5}$$

$$r_\theta^U(x, y) = \ell_\theta(x, y) - \widetilde{C}_{\mathcal{S}}(x, y), \tag{6}$$

where the frozen initialization π_0 anchors desirable behavior and the source-relative reward compares the target's support for a verified error with the aligned expert consensus. Let $\widehat{z}_0$ be the detached policy-anchored reference-point estimate used during optimization. The corresponding values are

$$v_\theta^{\mathrm{CGPO}}(x, y, z) = \begin{cases} \sigma\big(\beta(r_\theta^D(x, y) - \widehat{z}_0)\big), & z = D, \\ \sigma\big(\beta(\widehat{z}_0 - r_\theta^U(x, y))\big), & z = U. \end{cases}$$

This assignment protects desirable behavior with the target initialization, while source support prioritizes verified-error correction without overriding the labels. Alternative reference assignments are examined in Appendices B.5 and C.1. Holding the policy, anchor, and prompt shift fixed, let $a_U = \beta(r_\theta^U - \widehat{z}_0)$. The undesirable loss obeys

$$\frac{\partial \mathcal{L}_U}{\partial \widetilde{C}_{\mathcal{S}}} = -\beta\sigma(a_U)(1 - \sigma(a_U)) < 0.$$

Thus lower aligned expert support assigns a larger loss to a verifier-identified error, whereas high support yields a more conservative penalty. This establishes the monotonic ordering; Section 4.3 analyzes the resulting gradient magnitude.

4.2 Consensus Calibration and Stabilization

CGPO calibrates heterogeneous experts and stabilizes early optimization through three components.

Confidence-calibrated expert aggregation. Because neural confidence can be miscalibrated (Guo et al., 2017), a uniform mean ignores completion-specific expertise. For source q_i's tokeniza-

tion $y^{(i)} = (y_1^{(i)}, \ldots, y_{T_i}^{(i)})$, we compute predictive entropy

$$h_i(x, y) = \frac{1}{T_i} \sum_{t=1}^{T_i} \frac{-\sum_{v \in \mathcal{V}_i} q_i(v \mid x, y_{<t}^{(i)}) \log q_i(v \mid x, y_{<t}^{(i)})}{\log |\mathcal{V}_i|}.$$

The denominator places h_i in $[0, 1]$. We standardize it with calibration-split statistics

$$\bar{h}_i(x, y) = \mathrm{clip}\left(\frac{h_i(x, y) - \mu_i}{\max(\sigma_i, \varepsilon)}, -c, c \right).$$

Statistics may be task-conditioned. With temperature $\tau > 0$ and floor $0 \le \rho < 1/N$, the weights are

$$a_i(x, y) = \frac{\exp(-\bar{h}_i(x, y)/\tau)}{\sum_{j=1}^{N} \exp(-\bar{h}_j(x, y)/\tau)}, \tag{7}$$

$$w_i(x, y) = \rho + (1 - N\rho) a_i(x, y), \qquad 0 \le \rho < \frac{1}{N}. \tag{8}$$

Hence $w_i \ge 0$ and $\sum_i w_i = 1$: τ controls concentration, while ρ prevents any expert from being discarded. Entropy measures relative confidence, not correctness; all consensus quantities are computed offline and detached during target optimization.

Prompt-level source–target baseline alignment. Inspired by contextual calibration of prompt-dependent output shifts (Zhao et al., 2021), we correct additive score offsets between model families that remain after length normalization. For a fixed offline candidate set $\mathcal{Y}_x$ associated with prompt x, we therefore compute

$$B_{\pi_0}(x) = \frac{1}{|\mathcal{Y}_x|} \sum_{y' \in \mathcal{Y}_x} \ell_{\pi_0}(x, y'), \qquad B_{\mathcal{S}}(x) = \frac{1}{|\mathcal{Y}_x|} \sum_{y' \in \mathcal{Y}_x} C_{\mathcal{S}}(x, y'), \tag{9}$$

$$\delta_{\mathrm{BS}}(x) = B_{\pi_0}(x) - B_{\mathcal{S}}(x), \qquad \widetilde{C}_{\mathcal{S}}(x, y) = C_{\mathcal{S}}(x, y) + \delta_{\mathrm{BS}}(x). \tag{10}$$

This aligns the source-reference and initial-policy means, $|\mathcal{Y}_x|^{-1} \sum_{y' \in \mathcal{Y}_x} \widetilde{C}_{\mathcal{S}}(x, y') = B_{\pi_0}(x)$, while preserving all within-prompt source-score differences. Equivalently,

$$r_\theta^U(x, y) = \big[\ell_\theta(x, y) - B_{\pi_0}(x)\big] - \big[C_{\mathcal{S}}(x, y) - B_{\mathcal{S}}(x)\big],$$

so the undesirable reward compares within-prompt deviations rather than raw cross-model scores. CGPO applies the aligned source reference to undesirable completions while retaining π_0 as the anchor for desirable ones. All baseline quantities are computed offline and detached during optimization. Because the same scalar shift is added to every candidate for a prompt, alignment removes only the source–target location mismatch: it preserves the ordering and pairwise differences induced by source consensus. The target therefore receives the same relative expert guidance on which completions deserve stronger correction, but on a baseline comparable to its own scores.

D-first/U-warmup. At initialization $r_\theta^D = 0$ because $\pi_\theta = \pi_0$. Baseline alignment centers the undesirable reward over $\mathcal{Y}_x$, but individual source-relative rewards may remain nonzero. To avoid an abrupt source-driven update, CGPO activates the undesirable loss with

$$\alpha_U(t) = \mathrm{clip}\left(\frac{t - t_{\mathrm{start}}}{T_{\mathrm{warm}}}, 0, 1 \right),$$

for $T_{\mathrm{warm}} > 0$ (or $\alpha_U = 1$ without warmup). The U branch is inactive before t_{start} and then grows linearly to full weight; the D and U terms remain independent pointwise losses.

Unified objective. CGPO minimizes

$$\begin{aligned} \mathcal{L}_{\mathrm{CGPO}}(\theta; t) = {} & \lambda_D \mathbb{E}_{(x,y)\sim\mathcal{D}_D} \left[1 - \sigma\big(\beta(r_\theta^D(x, y) - \widehat{z}_0)\big) \right] \\ & + \alpha_U(t) \lambda_U \mathbb{E}_{(x,y)\sim\mathcal{D}_U} \left[1 - \sigma\big(\beta(\widehat{z}_0 - r_\theta^U(x, y))\big) \right], \end{aligned} \tag{11}$$

where $\lambda_D, \lambda_U \ge 0$ balance the labels. When $\widetilde{C}_{\mathcal{S}} = \ell_{\pi_0}$ and $\alpha_U = 1$, the objective reduces to the length-normalized KTO objective in Section 3.2; when all calibrated entropies are equal, $w_i = 1/N$. Thus CGPO adds completion-specific source guidance to pointwise preference learning. The objective separates three responsibilities: labels set the update direction, label-conditional references set completion-specific margins, and the curriculum controls when source-guided suppression becomes active. Unified notation and consensus properties are provided in Appendices B.1–B.2.

4.3 Gradient Analysis

We analyze how source consensus controls each pointwise update. Since $\widehat{z}_0$, ℓ_{π_0}, and $\widetilde{C}_{\mathcal{S}}$ are detached from the policy, define

$$m_D = r_\theta^D(x,y) - \widehat{z}_0, \qquad m_U = \widehat{z}_0 - r_\theta^U(x,y), \tag{12}$$

$$g_\beta(m) = \sigma(\beta m)\big(1-\sigma(\beta m)\big). \tag{13}$$

Direct differentiation of Eq. (11) gives

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\mathrm{CGPO}} = & -\mathbb{E}_{\mathcal{D}_D}\left[\lambda_D \beta g_\beta(m_D)\nabla_\theta \ell_\theta(x,y)\right] \\ & + \mathbb{E}_{\mathcal{D}_U}\left[\alpha_U(t)\lambda_U \beta g_\beta(m_U)\nabla_\theta \ell_\theta(x,y)\right].\end{aligned} \tag{14}$$

Gradient descent therefore promotes desirable and suppresses undesirable completions, with magnitude controlled by $g_\beta(m) \le 1/4$. The factor is largest near the value boundary and decreases as the signed margin becomes confident, so the reference changes optimization emphasis without reversing the label-imposed gradient direction.

To isolate source consensus, define the centered support gap

$$\widetilde{\Delta}_{\mathcal{S}}(x,y) = \ell_{\pi_0}(x,y) - \widetilde{C}_{\mathcal{S}}(x,y) = \ell_{\pi_0}(x,y) - C_{\mathcal{S}}(x,y) - \delta_{\mathrm{BS}}(x).$$

If m_U^0 is the undesirable margin obtained with the initial-policy reference alone, then $m_U = m_U^0 - \widetilde{\Delta}_{\mathcal{S}}$. Consensus preserves the suppression direction but changes its strength. It amplifies the update exactly when $|m_U^0 - \widetilde{\Delta}_{\mathcal{S}}| < |m_U^0|$, and attenuates it otherwise. Calibration weights source contributions, baseline alignment removes their prompt-common offset, and $\alpha_U(t)$ delays their effect. CGPO thereby prioritizes source-relevant corrections without changing the verifier-defined direction. Full gradient and scaling derivations are provided in Appendices B.3–B.4.

5 Experiments

5.1 Experimental Setup

Models. We use `microsoft/Phi-3.5-mini-instruct`, a 3.8B-parameter model, as the target (Abdin et al., 2024). Four sources—Qwen2.5-7B, Qwen2.5-7B-Instruct, Gemma-2-9B-IT, and Qwen2.5-Math-7B-Instruct—span base, instruction-tuned, and mathematics-specialized models (Qwen Team, 2024; Gemma Team, 2024; Yang et al., 2024). All participate in consensus; CGPO retains the target's architecture and inference cost.

Training data. We train on UltraFeedback, approximately 64K prompts with response-level feedback (Cui et al., 2024). We preserve each completion's assessment as an independent desirable or undesirable label. Sources score the same text with their own tokenizers, and their offline sequence scores form the calibrated consensus in Section 4.2. The same prompt-level candidate set estimates the initial-policy and source means; evaluation data remain disjoint.

Evaluation. We evaluate mathematics on GSM8K, MATH, and TheoremQA (Cobbe et al., 2021; Hendrycks et al., 2021b; Chen et al., 2023); general reasoning on BBH and MMLU (Suzgun et al., 2022; Hendrycks et al., 2021a); instruction following on IFEval (Zhou et al., 2023); and code generation on MBPP (Austin et al., 2021). Table 1 reports official percentage scores; Appendix A details prompting, inference, and scoring.

Baselines. We compare CGPO with the unmodified target, all four source models, supervised fine-tuning (SFT), and FuseLLM (Wan et al., 2024a). Preference-optimization baselines include SFT-DPO (Rafailov et al., 2023), SFT-KTO (Ethayarajh et al., 2024), WRPO (Yang et al., 2025), and InfiFPO (Gu et al., 2025). All trainable methods share the 3.8B target and evaluation pipeline; target and source models are reference-only and excluded from best/second-best ranking.

Implementation details. We use 48-GB NVIDIA RTX A6000 GPUs, a learning rate of $5e{-}7$, $\beta = 0.08$, and desirable/undesirable weights of $1.2/1.0$. Entropy temperature and weight floor are 0.8 and 0.02; the undesirable loss warms up over 100 steps.

5.2 MAIN RESULTS

Table 1: Main results across seven benchmarks (%). Category averages are unweighted. Best and second-best fusion/alignment results are bold and underlined; reference-only models are excluded. Size is the final parameter count.

		Math				General Reasoning			Instruction	Code	
Models	**Size**	**GSM8K**	**MATH**	**TheoremQA**	**Avg.**	**BBH**	**MMLU**	**Avg.**	**IFEval**	**MBPP**	**Overall Avg.**
Target Model											
Phi-3.5-mini	3.8B	81.05	47.2	23.7	50.7	67.6	69.5	68.6	49.2	68.5	58.1
Source Models											
Qwen2.5-7B-Base	7B	84.1	50.2	35.4	56.6	70.4	73.5	72.0	33.1	73.3	60.0
Gemma-2-9B-IT	9B	78.2	44.6	23.7	48.8	59.3	73.1	66.2	70.2	75.2	60.6
Qwen2.5-7B-Instruct	7B	90.5	72.5	36.7	66.6	70.6	73.4	72.0	71.2	78.9	70.5
Qwen2.5-7B-Math	7B	91.6	83.3	21.5	65.5	34.2	39.7	37.0	36.1	5.3	44.5
Model Fusion Methods											
FuseLLM	3.8B	83.4	48.6	26.3	52.8	69.4	68.3	68.9	<u>53.7</u>	72.2	60.3
Preference Optimization Methods											
SFT	3.8B	82.7	48.5	24.1	51.8	68.3	70.4	69.4	50.1	69.3	59.0
SFT-DPO	3.8B	83.0	49.7	25.5	52.7	69.1	69.7	69.4	51.6	70.5	59.9
SFT-KTO	3.8B	83.4	49.2	25.4	52.7	68.7	<u>70.7</u>	69.7	50.5	68.9	59.5
WRPO	3.8B	83.9	49.8	<u>30.1</u>	54.6	70.1	70.4	<u>70.3</u>	51.3	73.2	61.3
InfiFPO	3.8B	**84.8**	<u>50.6</u>	29.0	<u>54.8</u>	**70.4**	69.3	69.9	52.1	**75.3**	<u>61.7</u>
Our Method											
CGPO	3.8B	<u>84.2</u>	**53.9**	**32.8**	**57.0**	<u>70.2</u>	**73.1**	**71.7**	**54.3**	<u>74.7</u>	**63.3**

Table 2: Representative training efficiency. Gain/h is the seven-benchmark Overall Avg. gain over the target per GPU hour.

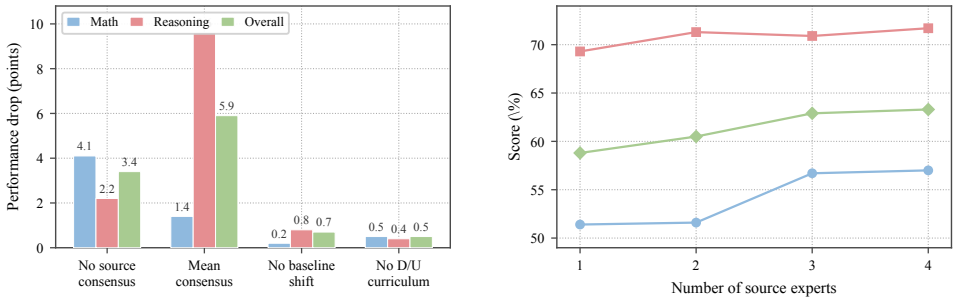
Method	Overall	GPU h	Gain/h
FuseLLM	60.3	162	0.014
WRPO	61.3	39	0.082
InfiFPO	61.7	40	0.090
CGPO	**63.3**	**36**	**0.144**

Broad and consistent improvements. CGPO leads all trainable methods overall, raising the target from 58.1 to 63.3 and surpassing the strongest baseline by 1.6 points. It improves every benchmark and ranks first on four and second on three. InfiFPO is slightly higher on GSM8K, BBH, and MBPP. Strong target performance on GSM8K and MBPP, together with limited source headroom on BBH, leaves less scope for consensus-guided undesirable correction. Pairwise fusion can also reinforce source-supported desirable responses, which CGPO anchors to π_0. Across the full suite, CGPO's label-conditional allocation yields the strongest overall result. The largest category gain is in mathematics ($+6.3$), where the specialist source offers substantial headroom, while simultaneous gains in the other categories show that transfer is not confined to that source's strongest domain.

Selective integration of heterogeneous expertise. The sources have sharply different profiles: Qwen2.5-7B-Math scores 83.3 on MATH but only 34.2 on BBH and 5.3 on MBPP, whereas the instruction-tuned sources are more balanced. CGPO improves all these domains rather than reproducing this trade-off. Verifier labels set the update direction, while calibrated consensus prioritizes undesirable corrections, exploiting relevant support while limiting weak-domain transfer. The uniform-consensus ablation further supports this interpretation.

Effective capability aggregation in a compact model. At 3.8B, CGPO exceeds three of four larger 7B–9B experts overall, consolidating their useful evidence into one independently deployable model without multi-model inference. The experts retain their full evaluation capacity, whereas CGPO is limited to 3.8B. Additional target-scale results are reported in Appendix C.2.

Training efficiency. CGPO reaches the best Overall Avg. in 36 GPU hours, fewer than WRPO (39), InfiFPO (40), and 4.5× fewer than FuseLLM (162). Its 0.144 points/GPU h is 1.6× InfiFPO and over 10× FuseLLM, so the gain does not rely on greater training cost. Relative to the preference-stage baselines, CGPO therefore improves both final quality and compute-normalized gain.



(a) Component ablations

(b) Number of source experts

Figure 3: Ablation results. (a) Score decreases from full CGPO (57.0 Math, 71.7 Reasoning, and 63.3 Overall). Mean consensus uses uniform source weights; no baseline shift uses unaligned consensus. (b) Performance versus source count; Overall is the unweighted mean over all seven benchmarks.

5.3 Ablation Study

To evaluate the contribution of each component in CGPO, we conduct an ablation study by comparing CGPO with its several degenerated versions.

Source guidance is the primary contributor. Removing source consensus lowers the overall average from 63.3 to 59.9 and returns performance to standard preference-optimization levels. The larger mathematics reduction is consistent with losing specialized evidence, identifying the source reference—rather than pointwise supervision alone—as the main transfer mechanism. Appendix D provides an illustrative qualitative example.

Calibration prevents negative transfer. Uniform averaging is more harmful than removing consensus entirely (-5.9 versus -3.4 overall). Equal weights let each source affect correction priority even in weak domains; the 9.6-point reasoning decrease is consistent with the mathematics specialist's weakness there. Although entropy is not a correctness estimator, completion-specific weighting is substantially more reliable than a uniform mean for deciding when each source should matter.

Baseline alignment and curriculum provide complementary stability. Baseline alignment preserves within-prompt source differences but removes the source–target offset that otherwise moves undesirable margins across the sigmoid. Its prompt-shared shift preserves candidate ordering while making the absolute margin scale comparable. D-first/U-warmup anchors desirable behavior before introducing source-guided suppression, reducing early branch competition. Their smaller effects identify both as stability mechanisms rather than primary sources of transferred knowledge.

Source-pool breadth and composition both matter. The overall average increases from 58.8 with one source to 63.3 with all four, showing that a broader pool supplies complementary evidence. Adding the third source sharply improves mathematics while slightly reducing reasoning, so the introduced expertise matters alongside source count. The four-source pool is best in both categories, indicating that broader coverage is most useful when calibration regulates completion-specific contributions.

6 Conclusions

We introduced Consensus-Guided Preference Optimization (CGPO), a preference-stage framework that fuses heterogeneous language models from independently labeled completions. CGPO combines verifier-defined directions with calibrated source consensus to prioritize pointwise corrections, avoiding response pairs, parameter merging, vocabulary alignment, and multi-model inference. Entropy calibration, baseline alignment, and D-first/U-warmup improve reliability. Across seven benchmarks, CGPO improves Phi-3.5 from 58.1 to 63.3 and outperforms all evaluated baselines. Ablations identify source guidance as primary and confirm the value of calibration and additional experts. Overall, CGPO compresses complementary source capabilities into one deployable model.

AI USE STATEMENT

In this work, we used generative AI tools solely to aid and polish the writing. Specifically, they were used for language editing to improve the grammar, clarity, concision, and presentation of author-written text. We did not use generative AI tools for literature retrieval or discovery, research ideation or execution, drafting substantive sections of the paper, generating synthetic datasets, or proving mathematical claims. The authors reviewed and edited all AI-assisted revisions and verified that they preserved the intended technical meaning. We take responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. UltraFeedback: Boosting language models with scaled AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*, 2024.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, 2024.

Yanggan Gu, Yuanyi Wang, Zhaoyi Yan, Yiming Zhang, Qi Zhou, Fei Wu, and Hongxia Yang. InfiFPO: Implicit model fusion via preference optimization in large language models. In *Advances in Neural Information Processing Systems*, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, 2021b.

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

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178. Association for Computational Linguistics, 2023.

Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. Binary classifier optimization for large language model alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1858–1872. Association for Computational Linguistics, 2025.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *International Conference on Learning Representations*, 2024.

Michael S. Matena and Colin A. Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, 2024.

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

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models. In *International Conference on Learning Representations*, 2024a.

Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. FuseChat: Knowledge fusion of chat models. *arXiv preprint arXiv:2408.07990*, 2024b.

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 *Advances in Neural Information Processing Systems*, volume 35, 2022.

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 *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Ziyi Yang, Fanqi Wan, Longguang Zhong, Tianyuan Shi, and Xiaojun Quan. Weighted-reward preference optimization for implicit model fusion. In *International Conference on Learning Representations*, 2025.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF: Rank responses to align language models with human feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706, 2021.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

APPENDIX

A DETAILED BENCHMARK AND EVALUATION SETUP

We evaluate every model with the same benchmark-specific prompts, demonstrations, inference procedure, and scorer. We use the standard evaluation split for each benchmark and do not use evaluation examples during training or model selection. Table 3 summarizes the protocol. The setting column reports the number of in-context demonstrations; CoT denotes that the prompt elicits a reasoning trace before the final answer. Greedy tasks use a single deterministic generation, whereas multiple-choice (MC) tasks rank the provided options by conditional log-likelihood and do not sample a free-form response.

Table 3: Evaluation protocol for the seven benchmarks in the main results. All generative evaluations use greedy decoding; MC denotes multiple choice.

Domain	Benchmark	Setting	Metric	Inference
Math	GSM8K	8-shot CoT	Exact match	Greedy
Math	MATH	0-shot CoT	Exact match	Greedy
Math	TheoremQA	5-shot	Answer-match accuracy	Greedy
Reasoning	BBH	3-shot	Normalized accuracy	MC log-likelihood
Reasoning	MMLU	5-shot	Accuracy	MC log-likelihood
Instruction	IFEval	0-shot	Prompt-level strict accuracy	Greedy
Code	MBPP	3-shot	pass@1	Greedy

A.1 MATHEMATICAL REASONING

GSM8K. GSM8K evaluates multi-step grade-school mathematical reasoning (Cobbe et al., 2021). We use the standard eight-shot chain-of-thought prompt (Wei et al., 2022) and greedy decoding. The final numerical answer is extracted and normalized according to the benchmark convention; a prediction is correct only when it exactly matches the reference answer after normalization. Intermediate reasoning is not scored separately.

MATH. MATH contains competition-level problems spanning several mathematical subjects and difficulty levels (Hendrycks et al., 2021b). We use zero-shot chain-of-thought prompting (Kojima et al., 2022) and greedy decoding. The evaluator extracts the final answer, applies the benchmark-specific normalization, and reports exact-match accuracy. The generated derivation may support answer production but does not itself contribute to the score.

TheoremQA. TheoremQA tests whether models can apply scientific and mathematical theorems to answer theorem-driven questions (Chen et al., 2023). We evaluate in the five-shot setting with greedy decoding. Predictions are parsed using the benchmark answer format and compared with the reference using its answer-matching rule; the reported metric is the fraction of examples answered correctly.

A.2 GENERAL REASONING

BBH. BIG-Bench Hard (BBH) collects challenging tasks that require algorithmic, symbolic, and logical reasoning (Suzgun et al., 2022). We use three in-context demonstrations. For each multiple-choice instance, the evaluator scores the candidate options by conditional log-likelihood and applies the benchmark's normalization before selecting the prediction. We report normalized accuracy aggregated over the benchmark.

MMLU. MMLU measures broad knowledge and reasoning across academic and professional subjects (Hendrycks et al., 2021a). We follow the standard five-shot protocol. Each answer option is scored by its conditional log-likelihood under the model, and the option with the highest score is selected. Accuracy is the proportion of questions for which this option equals the reference answer.

A.3 Instruction Following and Code Generation

IFEval. IFEval evaluates adherence to objectively verifiable natural-language instructions (Zhou et al., 2023). We use the zero-shot prompts and greedy decoding, then run the benchmark's deterministic instruction checkers. We report prompt-level strict accuracy: a prompt is counted as correct only if the response satisfies every verifiable instruction associated with that prompt.

MBPP. MBPP evaluates the synthesis of short Python programs from natural-language specifications (Austin et al., 2021). We use three-shot prompting and generate one solution per problem with greedy decoding. The generated program is executed against the benchmark test cases, and it contributes to pass@1 only when all tests pass.

A.4 Decoding and Score Aggregation

Greedy decoding removes sampling variance from GSM8K, MATH, TheoremQA, IFEval, and MBPP: each model produces exactly one completion for each example. BBH and MMLU are evaluated deterministically from option likelihoods. All per-benchmark scores in Table 1 are reported as percentages. The Math and General Reasoning columns are unweighted means over benchmarks within the corresponding category. Overall Avg. is the unweighted mean of all seven benchmark scores, not the mean of Math Avg. and Reasoning Avg. Target models, source models, and trained baselines are evaluated with the same protocol; no method-specific prompting or decoding adjustment is used.

B Additional Derivations and Properties of CGPO

This section collects notation used in the derivations and establishes results omitted from the main text. All log probabilities below are the completion-only, length-normalized quantities defined in Section 3.1.

B.1 Unified Label-Conditional Notation

Let $s_D = +1$ and $s_U = -1$, and define the label-conditional reference score and training multiplier as

$$b_z(x,y) = \begin{cases} \ell_{\pi_0}(x,y), & z = D, \\ \widetilde{C}_{\mathcal{S}}(x,y), & z = U, \end{cases} \qquad \gamma_z(t) = \begin{cases} 1, & z = D, \\ \alpha_U(t), & z = U. \end{cases} \tag{15}$$

Writing $r_\theta^z = \ell_\theta - b_z$, define the signed value margin and per-example loss as

$$m_z(x,y) = s_z\big(r_\theta^z(x,y) - \widehat{z}_0\big). \tag{16}$$

$$\mathcal{L}_z(x,y;t) = \gamma_z(t)\lambda_z\left[1 - \sigma(\beta m_z(x,y))\right]. \tag{17}$$

Taking the class-conditional expectation of Eq. (17) recovers Eq. (11). If $\widetilde{C}_{\mathcal{S}} = \ell_{\pi_0}$ and $\alpha_U(t) = 1$, both branches use the same reference and recover the corresponding length-normalized KTO objective under the same class-conditional aggregation (equivalently, with class priors absorbed into λ_D and λ_U).

B.2 Properties of Calibrated Source Consensus

Fix an example (x, y) and abbreviate $s_i = s_i(x,y)$, $a_i = a_i(x,y)$, and $w_i = w_i(x,y)$. For $\tau > 0$ and $0 \le \rho < 1/N$, the softmax weights satisfy $a_i > 0$ and $\sum_i a_i = 1$. It follows directly that

$$\sum_{i=1}^{N} w_i = N\rho + (1 - N\rho)\sum_{i=1}^{N} a_i = 1, \qquad \rho \le w_i \le 1 - (N-1)\rho. \tag{18}$$

Thus the calibrated consensus is a convex combination of source sequence scores and obeys

$$\min_i s_i \le C_{\mathcal{S}}(x,y) = \sum_i w_i s_i \le \max_i s_i. \tag{19}$$

These statements concern sequence scores; they do not require $C_{\mathcal{S}}$ to define a normalized probability distribution over completions.

The response of the weights to calibrated entropy is also explicit. Holding all other calibrated entropies fixed,

$$\frac{\partial a_i}{\partial \bar{h}_k} = -\frac{1}{\tau} a_i(\mathbb{I}[i=k] - a_k), \tag{20}$$

$$\frac{\partial w_i}{\partial \bar{h}_k} = -\frac{1-N\rho}{\tau} a_i(\mathbb{I}[i=k] - a_k). \tag{21}$$

In particular, for $N > 1$,

$$\frac{\partial w_i}{\partial \bar{h}_i} = -\frac{1-N\rho}{\tau} a_i(1-a_i) < 0, \tag{22}$$

so lowering an expert's calibrated entropy increases its own weight. If the scores s_i are held fixed, differentiation of the consensus further gives

$$\frac{\partial C_{\mathcal{S}}}{\partial \bar{h}_k} = -\frac{1-N\rho}{\tau} a_k \left(s_k - \sum_{i=1}^{N} a_i s_i \right). \tag{23}$$

Therefore confidence calibration increases the influence of a source only in relation to its sequence score; entropy itself is not treated as evidence of correctness. Finally, $\tau \to \infty$ yields $w_i \to 1/N$. If source k is the unique minimum-entropy source, then as $\tau \to 0^+$, $w_k \to 1-(N-1)\rho$ and $w_i \to \rho$ for $i \neq k$. All consensus quantities are precomputed and detached during policy optimization.

Baseline alignment changes only the prompt-level location of the consensus. Let $\Delta_{\mathcal{S}}(x,y) = \ell_{\pi_0}(x,y) - C_{\mathcal{S}}(x,y)$ and $\widetilde{\Delta}_{\mathcal{S}}(x,y) = \ell_{\pi_0}(x,y) - \widetilde{C}_{\mathcal{S}}(x,y)$. By construction,

$$\frac{1}{|\mathcal{Y}_x|} \sum_{y' \in \mathcal{Y}_x} \widetilde{C}_{\mathcal{S}}(x,y') = B_{\pi_0}(x), \tag{24}$$

$$\frac{1}{|\mathcal{Y}_x|} \sum_{y' \in \mathcal{Y}_x} \widetilde{\Delta}_{\mathcal{S}}(x,y') = 0. \tag{25}$$

Moreover, for any $y_1, y_2 \in \mathcal{Y}_x$,

$$\widetilde{\Delta}_{\mathcal{S}}(x,y_1) - \widetilde{\Delta}_{\mathcal{S}}(x,y_2) = \Delta_{\mathcal{S}}(x,y_1) - \Delta_{\mathcal{S}}(x,y_2), \tag{26}$$

so the shift removes a prompt-common offset without changing within-prompt comparisons.

B.3 Full Gradient Derivation

During a policy update, ℓ_{π_0}, $\widetilde{C}_{\mathcal{S}}$, $\widehat{z}_0$, $\gamma_z(t)$, and λ_z are independent of θ. This corresponds to frozen references, offline source scores, and a stop-gradient anchor. Define

$$g_\beta(m) = \sigma(\beta m)\big(1 - \sigma(\beta m)\big). \tag{27}$$

Since $\nabla_\theta r_\theta^z = \nabla_\theta \ell_\theta$ and $\nabla_\theta m_z = s_z \nabla_\theta \ell_\theta$, the chain rule applied to Eq. (17) gives

$$\nabla_\theta \mathcal{L}_z(x,y;t) = -\gamma_z(t)\lambda_z \beta s_z g_\beta(m_z) \nabla_\theta \ell_\theta(x,y), \tag{28}$$

$$\nabla_\theta \ell_\theta(x,y) = \frac{1}{|y|} \sum_{j=1}^{|y|} \nabla_\theta \log \pi_\theta(y_j \mid x, y_{<j}). \tag{29}$$

Taking the corresponding class-conditional expectations yields the gradient in Section 4.3. Since $0 < g_\beta(m) \leq 1/4$ for finite m, each per-example contribution satisfies

$$\|\nabla_\theta \mathcal{L}_z\| \leq \frac{\gamma_z(t)\lambda_z \beta}{4} \|\nabla_\theta \ell_\theta(x,y)\|. \tag{30}$$

B.4 Exact Consensus-Induced Gradient Scaling

Recall the initial-policy reward and centered source–target support gap

$$r_\theta^0 = \ell_\theta - \ell_{\pi_0}, \qquad \widetilde{\Delta}_{\mathcal{S}} = \ell_{\pi_0} - \widetilde{C}_{\mathcal{S}} = \Delta_{\mathcal{S}} - \delta_{\mathrm{BS}}. \tag{31}$$

For $m_U^0 = \widehat{z}_0 - r_\theta^0$, Section 4.3 gives $m_U = m_U^0 - \widetilde{\Delta}_{\mathcal{S}}$. Since

$$g_\beta(m) = \frac{1}{4\cosh^2(\beta m/2)}, \tag{32}$$

with all other factors fixed, the ratio between the source-guided and initial-reference gradient magnitudes is

$$\frac{g_\beta(m_U)}{g_\beta(m_U^0)} = \frac{\cosh^2(\beta m_U^0/2)}{\cosh^2(\beta(m_U^0 - \widetilde{\Delta}_{\mathcal{S}})/2)}. \tag{33}$$

Because g_β is even and strictly decreases with $|m|$ away from zero, the ratio in Eq. (33) exceeds one if and only if $m_U^0 \widetilde{\Delta}_{\mathcal{S}} > 0$ and $|\widetilde{\Delta}_{\mathcal{S}}| < 2|m_U^0|$. Equality leaves the magnitude unchanged; all other cases attenuate it.

B.5 Reference-Assignment Formulations

We isolate how the same baseline-aligned source reference is assigned to the two label branches while holding the training data, label weights, curriculum, and detached anchor fixed. We refer to the three resulting designs as *Policy-Anchored Reference*, *Consensus-Anchored Reference*, and *Label-Conditional Reference*, respectively. Their branch references are

$$\text{Policy-Anchored:} \quad b_D = b_U = \ell_{\pi_0}, \tag{34}$$

$$\text{Consensus-Anchored:} \quad b_D = b_U = \widetilde{C}_{\mathcal{S}}, \tag{35}$$

$$\text{Label-Conditional (CGPO):} \quad b_D = \ell_{\pi_0}, \qquad b_U = \widetilde{C}_{\mathcal{S}}. \tag{36}$$

Using r_θ^0 and $\widetilde{\Delta}_{\mathcal{S}}$ from Section B.4, their rewards are

$$r_\theta^{P,z} = r_\theta^0, \qquad r_\theta^{C,z} = r_\theta^0 + \widetilde{\Delta}_{\mathcal{S}}, \qquad r_\theta^{G,z} = r_\theta^0 + \mathbb{I}[z = U]\widetilde{\Delta}_{\mathcal{S}}. \tag{37}$$

Let $m_D^0 = r_\theta^0 - \widehat{z}_0$ and $m_U^0 = \widehat{z}_0 - r_\theta^0$. The corresponding signed margins satisfy

$$m_D^P = m_D^0, \qquad m_U^P = m_U^0, \tag{38}$$

$$m_D^C = m_D^0 + \widetilde{\Delta}_{\mathcal{S}}, \qquad m_U^C = m_U^0 - \widetilde{\Delta}_{\mathcal{S}}, \tag{39}$$

$$m_D^G = m_D^0, \qquad m_U^G = m_U^0 - \widetilde{\Delta}_{\mathcal{S}}. \tag{40}$$

Policy-Anchored Reference is a source-free control: both labels measure policy change relative to the frozen initialization. Consensus-Anchored Reference instead inserts the source–target shift into both branches, allowing consensus to affect the promotion of desirable completions as well as the suppression of undesirable ones. CGPO uses Label-Conditional Reference to separate these roles: the policy anchor preserves verified desirable behavior, while source consensus determines the value of verified errors.

Equations (37)– (40) isolate reference assignment. If the anchor estimator, source prior, row weights, or curriculum also changes, differences cannot be attributed to reference assignment alone. Mode-specific anchors leave the reward identities in Eq. (37) intact but add their anchor differences to the margin relations in Eq. (40).

C Additional Experiments

C.1 Reference-Assignment Ablation

We compare the three reference assignments formalized in Section B.5 while holding the training data, label weights, curriculum, and detached anchor fixed.

Table 4: Comparison of reference-assignment variants under the same training and evaluation protocol. All values are percentages. Math and General Reasoning averages follow Table 1. Overall Avg. is the unweighted mean of all seven benchmark scores, not the mean of Math Avg. and Reasoning Avg. The best result in each column is shown in bold.

Reference assignment	Math Avg.	Reasoning Avg.	Overall Avg.
Consensus-Anchored	56.9	71.2	63.1
Policy-Anchored	56.7	71.6	62.9
Label-Conditional (CGPO)	**57.0**	**71.7**	**63.3**

The two single-reference controls expose complementary failure modes. Policy-Anchored Reference preserves the target's desirable behavior but cannot use source evidence to prioritize undesirable corrections. Consensus-Anchored Reference adds that evidence to both branches; its higher mathematics score but lower reasoning score is consistent with specialist knowledge helping where the target is weak while perturbing verified behavior when applied indiscriminately. Label-Conditional Reference performs best in every aggregate by retaining the policy anchor for desirable completions and restricting source guidance to undesirable ones. Because the controls hold the remaining training choices fixed, the comparison supports CGPO's branch-specific assignment.

C.2 Target-Model Scale Robustness

We further test whether CGPO remains effective as the target model gains capacity. In addition to the 3.8B Phi-3.5-mini target used in the main experiments, we evaluate `microsoft/Phi-3-small-8k-instruct` (7B) and `microsoft/Phi-3-medium-4k-instruct` (14B) with the same seven-benchmark evaluation protocol (Abdin et al., 2024). Table 5 reports the corresponding capability-group and overall averages.

Table 5: Results with 7B and 14B target models. All values are percentages. Math and Reasoning averages are unweighted within each category. Overall Avg. is the unweighted mean of all seven benchmark scores, not the mean of Math Avg. and Reasoning Avg. The best and second-best results among fusion and alignment methods are shown in bold and underlined, respectively; the unmodified target is excluded from this ranking.

(a) Phi-3-small (7B)

Method	Math Avg.	Reasoning Avg.	Overall Avg.
Target Model	54.5	77.4	63.8
FuseLLM	56.1	77.6	65.0
SFT	55.5	78.0	64.5
SFT-DPO	56.1	77.9	65.2
SFT-KTO	56.0	78.2	65.5
WRPO	57.5	<u>78.7</u>	66.3
InfiFPO	<u>57.7</u>	78.4	<u>66.6</u>
CGPO	**59.3**	**79.6**	**67.8**

(b) Phi-3-medium (14B)

Method	Math Avg.	Reasoning Avg.	Overall Avg.
Target Model	56.4	79.7	66.1
FuseLLM	57.0	79.6	66.8
SFT	57.2	80.1	66.7
SFT-DPO	57.6	80.1	67.2
SFT-KTO	57.8	79.8	67.0
WRPO	58.4	<u>80.4</u>	67.6
InfiFPO	<u>58.8</u>	80.3	<u>67.7</u>
CGPO	**59.6**	**81.0**	**68.6**

CGPO achieves the best mathematics, reasoning, and overall averages with both larger targets, so its advantage is not confined to the 3.8B setting. Its target-relative overall gain decreases from 5.2 to 4.0 and 2.5 points as the target grows from 3.8B to 7B and 14B. A stronger target leaves fewer errors for source guidance to correct and narrows the capability gap from which consensus can draw, naturally reducing the available headroom. Nevertheless, CGPO still improves both capability groups at 14B and remains ahead of every compared fusion and alignment method. The mechanism is therefore robust over the evaluated scale range, although the diminishing gain does not imply scale invariance beyond it.

D Qualitative Case Study

Table 6 presents a MATH example (Hendrycks et al., 2021b) for which the original target model and its SFT variant are both incorrect, while CGPO reaches the correct answer. We report the complete generated solutions. The wording and calculations are preserved; only whitespace and mathematical typography are normalized.

Table 6: Complete model-generated solutions for a Level-4 number-theory problem from MATH. Incorrect calculations are retained exactly as generated.

Topic: Number Theory (MATH, Level 4)	
Problem	Find the remainder when 3^{1999} is divided by 13. **Reference answer:** 3.
Target (Phi-3.5-mini)	We will use Fermat's Little Theorem, which states that if p is a prime number, then for any integer a, $a^p \equiv a \pmod{p}$. In this case, $p = 13$ and $a = 3$. Fermat's Little Theorem tells us that $3^{13} \equiv 3 \pmod{13}$. Now, we want to find the remainder when 3^{1999} is divided by 13. First, let's find the remainder when 1999 is divided by 12 (since $13 - 1 = 12$). $1999 \equiv 3 \pmod{12}$ Now, we can rewrite 3^{1999} as $(3^{12})^{16} \times 3^3$. Using Fermat's Little Theorem, we know that $(3^{12})^{16} \equiv 3^{16} \pmod{13}$. $3^{16} \equiv 1 \pmod{13}$ because $3^{12} \equiv 1 \pmod{13}$ and raising both sides to the power of 2 gives us $3^{24} \equiv 1^2 \pmod{13}$, which simplifies to $3^{24} \equiv 1 \pmod{13}$. Now, we have $(3^{16})^{16} \times 3^3 \equiv 1^{16} \times 3^3 \pmod{13}$. This simplifies to $3^3 \equiv 27 \pmod{13}$. Finally, we find the remainder when 27 is divided by 13: $27 \equiv 1 \pmod{13}$ Final Answer: $\boxed{1}$

SFT	To find the remainder when 3^{1999} is divided by 13, we can use Fermat's Little Theorem. Fermat's Little Theorem states that if p is a prime number and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$. Here, $p = 13$ and $a = 3$. Since 3 is not divisible by 13, we can apply Fermat's Little Theorem: $3^{13-1} \equiv 3^{12} \equiv 1 \pmod{13}$ Now, we want to find $3^{1999} \pmod{13}$. We can express 1999 as $1999 = 12 \cdot 166 + 7$. Therefore, $3^{1999} = 3^{12 \cdot 166 + 7} = (3^{12})^{166} \cdot 3^7$ Using Fermat's Little Theorem, we know that $3^{12} \equiv 1 \pmod{13}$, so $(3^{12})^{166} \equiv 1^{166} \equiv 1 \pmod{13}$ Now, we need to find $3^7 \pmod{13}$. We can calculate this directly: $3^7 = 3 \cdot 3^6 = 3 \cdot (3^3)^2 = 3 \cdot 27^2 = 3 \cdot 7^2 = 3 \cdot 49 = 147$ Now, we find the remainder when 147 is divided by 13: $147 \equiv 8 \pmod{13}$ Therefore, $3^{1999} \equiv (3^{12})^{166} \cdot 3^7 \equiv 1 \cdot 8 \equiv 8 \pmod{13}$ The final answer is: $\boxed{8}$
CGPO	To find the remainder when 3^{1999} is divided by 13, we can use Fermat's Little Theorem. Fermat's Little Theorem states that if p is a prime number and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$. Here, $p = 13$ and $a = 3$. Since 3 is not divisible by 13, we can apply Fermat's Little Theorem: $3^{13-1} \equiv 3^{12} \equiv 1 \pmod{13}$ Now, we need to express 3^{1999} in terms of 3^{12}. We can write 1999 as $12 \times 166 + 7$: $3^{1999} = 3^{12 \times 166 + 7} = (3^{12})^{166} \cdot 3^7$ Using the result from Fermat's Little Theorem: $(3^{12})^{166} \cdot 3^7 \equiv 1^{166} \cdot 3^7 \pmod{13}$ Since $1^{166} = 1$, we have: $1 \cdot 3^7 \equiv 3^7 \pmod{13}$ Now, we need to calculate $3^7 \pmod{13}$: $3^7 = 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 2187$ To find $2187 \pmod{13}$, we can perform the division: $2187 = 168 \cdot 13 + 3$ Therefore, $2187 \equiv 3 \pmod{13}$. So, the remainder when 3^{1999} is divided by 13 is: $3^{1999} \equiv 3 \pmod{13}$ Final Answer: $\boxed{3}$

All three models identify Fermat's little theorem as the appropriate tool, so the difference is not theorem recall but the reliability with which the theorem is executed. The target immediately makes the incorrect reduction $1999 \equiv 3 \pmod{12}$ instead of $1999 \equiv 7 \pmod{12}$ and subsequently applies inconsistent exponent manipulations. SFT repairs this high-level step by writing $1999 = 12 \cdot 166 + 7$, but its residual computation remains unstable: it replaces 27^2 with 7^2 without justification and then asserts $147 \equiv 8 \pmod{13}$, although the remainder is 4. Thus SFT learns the surface structure of a valid solution while retaining locally plausible but incorrect arithmetic transitions.

CGPO corrects both levels of failure. It preserves the valid decomposition $1999 = 12 \cdot 166 + 7$, applies $3^{12} \equiv 1 \pmod{13}$ without altering the residual exponent, and verifies the final computation through the explicit identity $2187 = 168 \cdot 13 + 3$. The resulting derivation is internally consistent from theorem selection to the final remainder. This behavior illustrates the correction encouraged by CGPO: pointwise verifier labels identify incorrect completions, while source consensus places greater pressure on errors unsupported by multiple sources. In this example, the result is not merely a different final answer but a coherent sequence of justified intermediate steps.