HARMONY IN DIVERSITY: MULTI-DOMAIN CONTRASTIVE POLICY OPTIMIZATION FOR LARGE REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training via Reinforcement Learning (RL) has enabled Large Reasoning Models (LRMs) to achieve strong performance in individual domain. However, real-world applications increasingly require general-purpose reasoners rendering strong performance across diverse domains. Mixed-domain post-training aims to achieve this goal by jointly training with mixed domain data, but this often induces capability compromise and degradation among different domains. Existing methods attribute this performance degradation to harmful cross-domain interactions and propose various strategies to mitigate them, but these strategies may also impede beneficial knowledge sharing across domains and in turn fail to match or surpass single-domain performance. To address this problem, we propose **M**ulti-domain **C**ontrastive **P**olicy **O**ptimization (MCPO), which uses contrastive learning to utilize both positive and negative cross-domain interactions for knowledge sharing and competition. Specifically, we partition each rollout generated by LRMs according to its underlying reasoning structures and use reasoning segments to capture these structures. We thus formulate positive and negative pairs of reasoning segments as mutually augmented examples, which provide supportive and competing signals for knowledge sharing. Subsequently, we design complementary contrastive objectives for cross-domain knowledge sharing and intra-domain knowledge consolidation, targeting compatibility across domains and discriminability within each domain to form a harmonious reasoning space. Experimental results across a broad range of domains show that MCPO alleviates performance degradation caused by mixed-domain training and outperforms single-domain training in most cases.

"Every culture cherishes its own beauty, and appreciates the beauty of others; When all beauties blend in harmony, the world achieves universal unity."

— Xiaotong Fei

1 INTRODUCTION

Recent advances in post-training have enabled large reasoning models (LRMs) to achieve strong performance in a certain target domain after training on domain-specific data (Shao et al., 2024; Fan et al., 2025; Rong et al., 2025). However, since real-world applications often require general-purpose reasoners which are simultaneously applicable to multiple domains (Wei et al., 2026; Bikkasani, 2025), it is increasingly important for LRMs to achieve strong performance across diverse domains. A common approach for achieving this goal is to post-train a single LRM on mixed-domain data, so that the performance across all domains can be improved. In practice, however, such mixed-domain training not only fails to improve performance across all domains simultaneously, but also induces capability trade-offs and even degradation (Wu et al., 2026; Cai et al., 2026; Ramesh et al., 2026; Liang et al., 2026). Specifically, the interactions among the different heterogeneous domains lead to the performance gains in some domains coming at the expense of the performance sacrifice of others.

Existing works attribute this issue to negative interactions during the learning of knowledge from different domains, and they propose to mitigate them by isolating learning directions (Yu et al., 2020; Liu et al., 2021; Cai et al., 2026; Liu et al., 2026) and balancing learning progress (Chen et al., 2018; Ramesh et al., 2026; Liang et al., 2026) among various domains. In most cases, these methods

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.

treat cross-domain interactions as harmful and develop various strategies to suppress or avoid them. However, indiscriminately mitigating cross-domain interactions may also impede the sharing of potentially beneficial knowledge across domains. Therefore, we argue that cross-domain interactions are not always negative which lead to conflicts or interference. They may instead contain some positive interactions which carry supportive reasoning structures and facilitate knowledge transfer between domains. For example, mathematical reasoning and code generation can rely on shared planning abilities, while mathematical formulas and programming syntax remain domain-specific.

Motivated by this observation, we utilize cross-domain interactions in two ways: pairs of reasoning structures whose interactions facilitate knowledge transfer are treated as positive pairs to provide supportive knowledge, whereas those whose interactions induce conflicts or interference are treated as negative pairs to provide competing knowledge. Specifically, to capture these reasoning structures, we partition each rollout generated by LRMs into reasoning segments through reasoning segmentation (Ai et al., 2026; Guo et al., 2025b), and treat these segments as the basic units for modeling cross-domain interactions. Therefore, we formulate pairs of reasoning segments that provide either supportive or competing knowledge as *mutually augmented examples* for contrastive learning (Chen et al., 2020; He et al., 2020; van den Oord et al., 2018; Poole et al., 2019), thereby transforming cross-domain interactions into supportive and competing signals for knowledge sharing during policy optimization.

Based on this formulation, we propose **Multi-domain Contrastive Policy Optimization (MCPO)**, a multi-domain post-training framework that adopts contrastive learning to transform positive and negative cross-domain interactions into supportive and competing signals for knowledge sharing. Specifically, at each optimization step, MCPO partitions each rollout into reasoning segments and applies a gradient variation test to determine whether updating on one segment facilitates or interferes with learning on another. The resulting positive and negative segment pairs construct mutually augmented examples carrying supportive and competing knowledge for knowledge sharing. These examples support two complementary contrastive objectives: **(i) cross-domain knowledge sharing**, which reinforces supportive knowledge from other domains to connect compatible reasoning structures; and **(ii) intra-domain knowledge consolidation**, which aligns supportive knowledge within the same domain to construct a discriminative reasoning space. Accordingly, MCPO jointly optimizes the contrastive objectives with GRPO (Shao et al., 2024) to achieve knowledge sharing across domains and then form a harmonious reasoning space.

Compared with traditional multi-domain post-training based on GRPO (Shao et al., 2024), MCPO offers three advantages. First, it unifies cross-domain knowledge sharing and intra-domain knowledge consolidation in a harmonious reasoning space, where transferable reasoning structures can improve reasoning performance across and within domains. Second, it fully exploits the underlying reasoning structures captured by reasoning segments, and enables supportive and competing knowledge to be coordinated across domains. Third, it models negative interactions as competing knowledge rather than indiscriminately isolating them, which allows diverse reasoning structures to enhance policy exploration. Extensive experiments across a broad range of domains, *e.g.*, Math, Code, Science, ToolUse, and Safety, show that MCPO effectively exploits positive and negative cross-domain interaction signals to construct a harmonious reasoning space. It also alleviates the performance degradation caused by multi-domain training and outperforms single-domain training in most cases.

2 Problem Formulation

2.1 Multi-domain Reinforcement Learning

Let $d \in \{1, \dots, D\}$ index a reasoning domain with prompt distribution $\mathcal{D}_d$. For a prompt $p \sim \mathcal{D}_d$, the domain index determines the verifier used to score its response. Multi-domain reinforcement learning optimizes a shared policy $\pi_{\boldsymbol{\theta}}$ over a mixture of domain-specific objectives:

$$\max_{\boldsymbol{\theta}} \ \mathcal{J}(\boldsymbol{\theta}) = \sum_{d=1}^{D} \alpha_d \mathbb{E}_{p\sim\mathcal{D}_d,\, y\sim\pi_{\boldsymbol{\theta}}(\cdot|p)} \big[R_d(p, y)\big], \tag{1}$$

where $\alpha_d > 0$ is the domain sampling weight with $\sum_{d=1}^{D} \alpha_d = 1$, and R_d is the domain-specific verifier. For example, R_d may return an exact-answer correctness score for a mathematics problem, a pass rate for a code problem, or a success score for tool use or safety. Joint optimization on mixed-domain data allows updates from different domains to reinforce or interfere with each other.

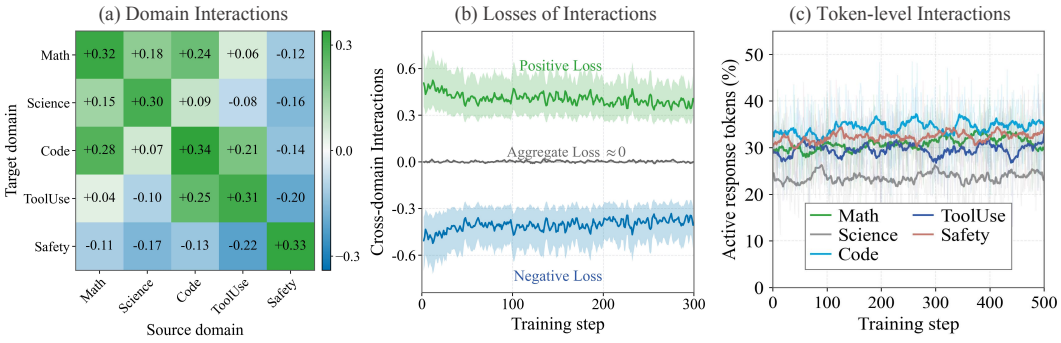

Figure 1: **Motivation for MCPO.** (a) Directed interactions contain both positive and negative domain relations. (b) The corresponding signed losses remain sizable while their aggregate is close to zero. (c) Active response-token ratios vary across domains, indicating domain-specific optimization signals.

2.2 Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) improves a policy by comparing candidate responses to the same prompt and can be adapted to multi-domain RL.

At each mixed-domain training step, a batch of B prompts $\mathcal{B} = \{p_b\}_{b=1}^{B}$ is sampled from the domain mixture, where each p_b belongs to a domain $d(p_b)$. For every prompt $p \in \mathcal{B}$, the old policy $\pi_{\boldsymbol{\theta}_{\text{old}}}$ generates a rollout group $\mathcal{Y}(p) = \{y_{p,g}\}_{g=1}^{G}$ of G responses, indexed by $g = 1, \ldots, G$. The domain verifier assigns each rollout a scalar reward $r_{p,g} = R_{d(p)}(p, y_{p,g})$. We use $y_{p,g,t}$ for the t-th token of $y_{p,g}$ and $y_{p,g,<t}$ for its preceding prefix. GRPO computes the group mean $\mu_p = \frac{1}{G}\sum_{g=1}^{G} r_{p,g}$, the group variance $\sigma_p^2 = \frac{1}{G}\sum_{g=1}^{G}(r_{p,g} - \mu_p)^2$, and the normalized advantage $A_{p,g} = \frac{r_{p,g}-\mu_p}{\sigma_p+\epsilon}$, where $\epsilon > 0$ is a numerical stabilizer. The token importance ratio is $\rho_{p,g,t} = \frac{\pi_{\boldsymbol{\theta}}(y_{p,g,t}|p,y_{p,g,<t})}{\pi_{\boldsymbol{\theta}_{\text{old}}}(y_{p,g,t}|p,y_{p,g,<t})}$. Let $N_{\mathcal{B}} = \sum_{p\in\mathcal{B}}\sum_{g=1}^{G}|y_{p,g}|$ denote the number of valid response tokens in the mixed batch. Under token-mean aggregation, GRPO minimizes

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{N_{\mathcal{B}}}\sum_{p\in\mathcal{B}}\sum_{g=1}^{G}\sum_{t=1}^{|y_{p,g}|}\min(\rho_{p,g,t}A_{p,g}, \text{clip}(\rho_{p,g,t}, 1-\varepsilon, 1+\varepsilon)A_{p,g})\,. \tag{2}$$

Here ε is the ratio-clipping parameter. Each advantage of the response is normalized within its prompt group. Mixed-domain training aggregates the prompt-local losses to update the shared policy.

2.3 Motivation: Treating Multi-domain Interactions Heterogeneously

GRPO constructs advantages independently within each prompt, then aggregates prompt-local gradients into a shared-policy update. Let p be a target prompt, p' a prompt in batch $\mathcal{B}$, $\mathcal{L}_p$ the sampled GRPO loss for p, and $\boldsymbol{g}_p = \nabla_{\boldsymbol{\theta}}\mathcal{L}_p$. For normalized weights $w_{p'} \geq 0$ with $\sum_{p'\in\mathcal{B}} w_{p'} = 1$, the update $\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta\sum_{p'\in\mathcal{B}} w_{p'}\boldsymbol{g}_{p'}$ gives the following first-order expansion under local smoothness:

$$\mathcal{L}_p(\boldsymbol{\theta}^+) - \mathcal{L}_p(\boldsymbol{\theta}) = -\eta w_p\|\boldsymbol{g}_p\|_2^2 - \eta\sum_{\substack{p'\in\mathcal{B}\\ p'\neq p}} w_{p'}\boldsymbol{g}_p^\top\boldsymbol{g}_{p'} + O(\eta^2). \tag{3}$$

The first term is the remaining descent contribution from the target prompt itself after all cross-prompt interactions are excluded; the second aggregates the effects of all other prompt gradients on the target loss. When $d(p') \neq d(p)$, these effects are cross-domain interactions: positive inner products decrease the target loss to first order, whereas negative ones increase it. Furthermore, the directed matrix in Fig. 1(a) shows that interactions occur both within and across domains, with their signs and strengths varying across domain pairs.

Existing methods act on such aggregates at the level of domain gradients. One line of work suppresses or projects away interactions to isolate learning directions (Yu et al., 2020; Liu et al., 2021; Cai et al., 2026), and another adaptively adjusts domain weights ζ_d to balance learning progress (Kendall et al., 2018; Chen et al., 2018; Ramesh et al., 2026). Neither explicitly resolves supportive and competing relations between individual reasoning structures, so transferable knowledge carried by

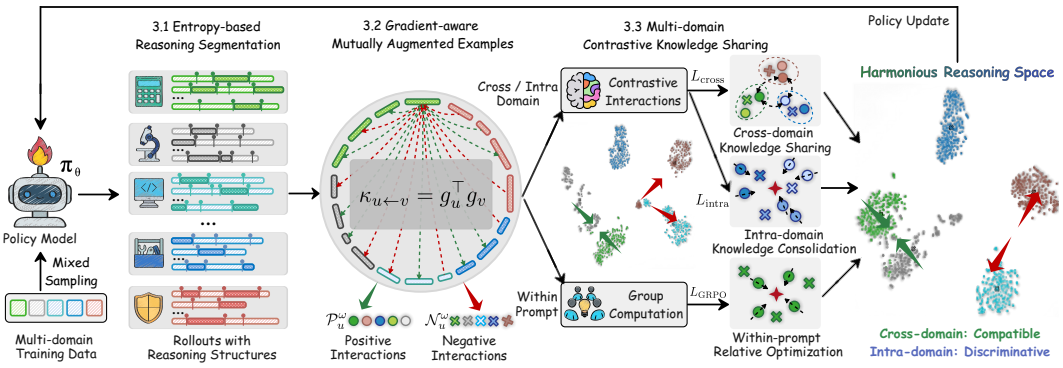

Figure 2: **Overview of MCPO.** MCPO adopts contrastive learning to transform positive and negative cross-domain interactions into mutually augmented examples with supportive and competing signals of knowledge sharing. It therefore induces a harmonious reasoning space with cross-domain compatibility and intra-domain discriminability.

positive relations may be attenuated alongside interference from negative ones. Curvature-guided second-order corrections (Liang et al., 2026) enter at $O(\eta^2)$ and vanish one order faster than the $O(\eta)$ interaction as $\eta \to 0$, leaving its signed composition equally unresolved.

We instead resolve the signed first-order interactions before aggregation. Consistent with recent analyses of multi-task LLM reinforcement learning (Zhu et al., 2026), our five-domain diagnostics show that the aggregate cross-prompt interaction is close to zero, as illustrated by the signed loss traces in Fig. 1(b). Each pairwise inner product $\boldsymbol{g}_p^\top \boldsymbol{g}_{p'}$ in Eq. (3) carries its own sign, and grouping the interaction term by sign gives

$$\sum_{p' \neq p} w_{p'} \boldsymbol{g}_p^\top \boldsymbol{g}_{p'} = \underbrace{\sum\nolimits_{p' \neq p} w_{p'} [\boldsymbol{g}_p^\top \boldsymbol{g}_{p'}]_+}_{M_p^+} - \underbrace{\sum\nolimits_{p' \neq p} w_{p'} [-\boldsymbol{g}_p^\top \boldsymbol{g}_{p'}]_+}_{M_p^-}, \tag{4}$$

where $[z]_+ = \max(z, 0)$. A near-zero aggregate therefore admits two explanations. Either the pairwise interactions are weak, so $M_p^+ \approx M_p^- \approx 0$, or supportive and competing interactions cancel each other, so M_p^+ and M_p^- are substantial but nearly equal. The aggregate retains only their difference $M_p^+ - M_p^-$ and cannot distinguish these cases or separately exploit positive relations for transfer and negative relations for competition.

Overall, the decomposition in Eq. (4) separates the aggregate into positive and negative interaction signals, while active response tokens in Fig. 1(c) and interaction dynamics in Appendix D.1 further show that rollouts generated by LRMs carry rich and domain-specific signals throughout training.

3 MCPO: Multi-domain Contrastive Policy Optimization

In this section, we propose **MCPO** (**M**ulti-domain **C**ontrastive **P**olicy **O**ptimization), a framework that transforms positive and negative cross-domain interactions into supportive and competing signals for knowledge sharing during the multi-domain policy optimization.

As shown in Figure 2, MCPO first divides each rollout into reasoning segments (Section 3.1). It then applies a gradient variation test to determine whether the update induced by one segment positively or negatively affects the learning of another (Section 3.2). Based on the test results, mutually augmented examples are identified as pairs of reasoning segments with obvious positive and negative interactions. Subsequently, MCPO designs two contrastive objectives (Section 3.3) for cross-domain knowledge sharing and intra-domain knowledge consolidation to form a harmonious reasoning space.

3.1 Entropy-based Reasoning Segmentation

We regard fine-grained reasoning structures as the basic units for utilizing cross-domain interactions for LRMs. Intuitively, a rollout from an LRM contains a sequence of diverse reasoning structures, such as planning, derivation, and verification, whose interactions with structures in other rollouts differ in sign and strength. To utilize these interactions, we should identify the reasoning structure in

each rollout, since their opposite effects cancel once the whole response is aggregated, as analyzed in Section 2.3. Formally, we use entropy-based segmentation (Ai et al., 2026; Guo et al., 2025b) to establish boundaries and then capture the reasoning structures.

Specifically, for a response $y = (y_1, y_2, \ldots, y_T)$ generated for prompt p, the entropy of predictive distribution for each token under the rollout policy and its response-relative threshold are defined as

$$H_t = -\sum\nolimits_{a\in\mathcal{V}} \pi_{\boldsymbol{\theta}_{\text{old}}}(a \mid p, y_{<t}) \log \pi_{\boldsymbol{\theta}_{\text{old}}}(a \mid p, y_{<t}), \quad \text{and} \quad \tau_y = Q_{q_H}\left(\{H_j\}_{j=1}^{T-1}\right), \tag{5}$$

respectively. Here, Q_{q_H} is the within-response quantile operator, $q_H \in (0, 1)$, and $\mathcal{V}$ is the vocabulary. Tokens with high predictive entropy are critical forks that steer the model towards different reasoning paths (Cui et al., 2025), so positions $t \in \{1, \ldots, T-1\}$ with $H_t \geq \tau_y$ are selected as candidate boundaries. The threshold takes a within-response quantile because the overall uncertainty level differs across prompts and domains, and a fixed global cut would segment confident and uncertain responses unevenly, whereas the quantile adapts the boundary criterion to each rollout. We defer the filtering rule to Appendix B.2. The ordered boundaries $0 = b_0 < b_1 < \cdots < b_{K_y} = T$ induce

$$s^k = y_{b_{k-1}+1:b_k}, \qquad y = s^1 \oplus \cdots \oplus s^{K_y}, \qquad \mathcal{S}(y) = (s^1, \ldots, s^{K_y}). \tag{6}$$

Let $\boldsymbol{g}_t$ be the gradient contribution of token t. For every segment s, $\boldsymbol{g}_s = \sum_{t\in s} \boldsymbol{g}_t$, where the term for each token is conditioned on the original prompt and its complete prefix. Because the segments are disjoint and they cover the whole response, the sum of their gradient exactly recovers the response gradient, as proved in Appendix B.1. Subsequently, the partition exposes the local units at which supportive and competing knowledge can be distinguished while preserving their trajectory context. These units supply the candidates for the construction of mutually augmented examples in our MCPO.

3.2 Gradient-aware Mutually Augmented Examples

We construct *mutually augmented examples* from pairs of reasoning segments with obvious positive and negative interactions, and leverage their gradient interaction to assign the role of supportive or competing knowledge. Here, a pair is positive when learning from its source facilitates learning from its target, and negative when the source interferes with the target. For example, a mathematical planning segment can support a coding planning segment when the parameter update induced by the former decreases the loss of the latter in the gradient of the origin rollout.

Formally, we quantify interactions with gradient variation tests, which capture the direction through the first-order loss variation induced by a gradient descent step on the source segment. Let $u \in \mathcal{S}(y_{p,g})$ and $v \in \mathcal{S}(y_{p',g'})$ denote a target and a source segment with segment gradients $\boldsymbol{g}_u$ and $\boldsymbol{g}_v$ from Section 3.1, and let ℓ_u denote the loss of target segment with $\boldsymbol{g}_u = \nabla_{\boldsymbol{\theta}} \ell_u$. Under the loss minimization framework, v induces the virtual update $\boldsymbol{\theta} - \eta \boldsymbol{g}_v$, and the test measures its effect on the target segment

$$\Delta_{u\leftarrow v}(\eta) = \ell_u(\boldsymbol{\theta} - \eta \boldsymbol{g}_v) - \ell_u(\boldsymbol{\theta}) = -\eta \boldsymbol{g}_u^\top \boldsymbol{g}_v + O\left(\eta^2 \|\boldsymbol{g}_v\|_2^2\right). \tag{7}$$

At a locally smooth point and for sufficiently small η, $\boldsymbol{g}_u^\top \boldsymbol{g}_v > 0$ implies $\Delta_{u\leftarrow v} < 0$, so learning from v reduces the loss of u and forms a positive relation, whereas $\boldsymbol{g}_u^\top \boldsymbol{g}_v < 0$ implies $\Delta_{u\leftarrow v} > 0$, which means the two segments compete with each other. The test therefore assigns a sign to each segment pair before positive and negative effects cancel in the prompt-level interaction of Eq. (3), and the virtual update introduces no additional optimizer step. Because segment gradients are additive, the interaction score expands over their token contributions as

$$\kappa_{u\leftarrow v} = \boldsymbol{g}_u^\top \boldsymbol{g}_v = \left(\sum\nolimits_{t\in u} \boldsymbol{g}_t\right)^\top \left(\sum\nolimits_{r\in v} \boldsymbol{g}_r\right) = \sum_{t\in u} \sum_{r\in v} \boldsymbol{g}_t^\top \boldsymbol{g}_r. \tag{8}$$

The score sums all token-pair gradient interactions between u and v and is precisely the inner product governing the first-order variation in Eq. (7), whereas its sign identifies each pair as positive or negative. In practice, $\kappa_{u\leftarrow v}$ is estimated by a lightweight gradient proxy, derived in Appendix B.4.

Subsequently, we denote $p(u)$ and $d(u)$ as the original prompt of segment u and the domain of that prompt, respectively. For each target u, let $\mathcal{C}_u^{\text{cross}}$ collect source segments from other domains and $\mathcal{C}_u^{\text{intra}}$ those from other prompts in the same domain. Let $\text{Top}_{q_R}(\mathcal{X}; f)$ denote the operator that returns the $\lceil q_R|\mathcal{X}|\rceil$ elements of a nonempty set $\mathcal{X}$ with the largest values of f. Here, $q_R \in (0, 1]$ is

the relation retention ratio, and we set $\mathrm{Top}_{q_R}(\emptyset; f) = \emptyset$. MCPO then filters candidates by their signs and retains the top-q_R fraction on each side

$$\begin{aligned} \mathcal{P}_u^\omega &= \mathrm{Top}_{q_R}(\{v \in \mathcal{C}_u^\omega : \kappa_{u\leftarrow v} > 0\}; \kappa_{u\leftarrow v}), \\ \mathcal{N}_u^\omega &= \mathrm{Top}_{q_R}(\{v \in \mathcal{C}_u^\omega : \kappa_{u\leftarrow v} < 0\}; -\kappa_{u\leftarrow v}), \qquad \omega \in \{\mathrm{cross}, \mathrm{intra}\}. \end{aligned} \tag{9}$$

Accordingly, the selected positive and negative pairs form mutually augmented examples, which carry explicit signs and strengths that can serve the contrastive objectives in Section 3.3.

3.3 Multi-domain Contrastive Knowledge Sharing

Based on the mutually augmented examples discovered above, we now transform the cross-domain interactions into supportive and competing signals for knowledge sharing through contrastive learning.

Specifically, for each scope $\omega \in \{\mathrm{cross}, \mathrm{intra}\}$, we define $\mathcal{A}_\omega = \{u : \mathcal{P}_u^\omega \neq \emptyset, \mathcal{N}_u^\omega \neq \emptyset\}$ as the targets with both relation types. We use $\tau_c > 0$ to represent the temperature in contrastive learning. MCPO then instantiates two objectives as contrastive losses over mutually augmented examples, one targeting compatibility across domains and the other discriminability within each domain.

Cross-domain knowledge sharing. Cross-domain candidates come from two distinct domains. The corresponding objective that facilitates knowledge sharing between them is given by

$$\mathcal{L}_{\mathrm{cross}} = -\frac{1}{|\mathcal{A}_{\mathrm{cross}}|} \sum\nolimits_{u \in \mathcal{A}_{\mathrm{cross}}} \log \frac{\sum_{v \in \mathcal{P}_u^{\mathrm{cross}}} \exp(\kappa_{u\leftarrow v}/\tau_c)}{\sum_{v \in \mathcal{P}_u^{\mathrm{cross}} \cup \mathcal{N}_u^{\mathrm{cross}}} \exp(\kappa_{u\leftarrow v}/\tau_c)}. \tag{10}$$

The positive numerator retains complementary domain-specific realizations of compatible reasoning structures, while the denominator contrasts them with competing cross-domain updates. Thus, Eq. (10) strengthens supportive reasoning interactions across domain boundaries without merging their distinct realizations, while its negatives keep opposed cross-domain updates competing.

Intra-domain knowledge consolidation. Intra-domain candidates come from distinct prompts in same domain capturing recurring reasoning structures. MCPO consolidates these relations through

$$\mathcal{L}_{\mathrm{intra}} = -\frac{1}{|\mathcal{A}_{\mathrm{intra}}|} \sum\nolimits_{u \in \mathcal{A}_{\mathrm{intra}}} \log \frac{\sum_{v \in \mathcal{P}_u^{\mathrm{intra}}} \exp(\kappa_{u\leftarrow v}/\tau_c)}{\sum_{v \in \mathcal{P}_u^{\mathrm{intra}} \cup \mathcal{N}_u^{\mathrm{intra}}} \exp(\kappa_{u\leftarrow v}/\tau_c)}. \tag{11}$$

Here, the positives reinforce discriminative reasoning structures recurring within the domain, whereas the negatives separate locally opposed decisions. Keeping Eq. (11) distinct from Eq. (10) prevents either scope from subsuming the other under heterogeneous sampling.

MCPO subsequently combines the two scope losses with prompt-local GRPO through

$$\mathcal{L}_{\mathrm{total}}(\boldsymbol{\theta}) = \mathcal{L}_{\mathrm{GRPO}}(\boldsymbol{\theta}) + \lambda_{\mathrm{cl}} \underbrace{\left(\lambda_{\mathrm{cross}} \mathcal{L}_{\mathrm{cross}} + \lambda_{\mathrm{intra}} \mathcal{L}_{\mathrm{intra}}\right)}_{\mathcal{L}_{\mathrm{MCPO}}}. \tag{12}$$

Here, λ_{cross} and λ_{intra} balance cross-domain knowledge sharing and intra-domain knowledge consolidation, while λ_{cl} controls their overall strength relative to GRPO (Shao et al., 2024). Cross-domain positive pairs share compatible reasoning structures, intra-domain positive pairs consolidate their recurring variants, and negative examples maintain contrastive competition between opposed updates. The same rollout batch thus supports within-prompt GRPO and segment-level knowledge sharing for a harmonious reasoning space. Negative pairs simultaneously participate as competing examples rather than being discarded; their effect on policy exploration is examined in Section 4.3.

4 Experiments

In this section, we conduct extensive experiments on multi-domain RL and provide detailed algorithmic analyses of our proposed MCPO.

4.1 Experimental Setup

In this part, we provide the detailed experimental setup, including backbone models, baseline methods, tasks, datasets and benchmarks, evaluation metrics, and implementation details.

Table 1: Accuracies of the adopted methods on nine benchmarks. *Avg.* is the unweighted mean over the nine benchmarks. The best and second-best results within each backbone model are shown in bold and underlined, respectively.

Method	Math			Code		Science		ToolUse	Safety	Overall
	AMC23	AIME24	AIME25	LCBv5	LCBv6	GPQA-D	SciBench	Toolpaca	Beaver	*Avg.*
Qwen3-4B										
Base Model (Team, 2025)	71.17	36.04	33.75	14.43	19.54	35.45	27.01	45.83	58.40	37.96
GRPO (Shao et al., 2024)	78.81	48.02	39.10	25.79	27.91	43.95	33.04	56.73	78.19	47.95
DAPO (Yu et al., 2025)	80.83	53.56	40.34	25.88	27.93	46.37	34.86	55.29	78.08	49.24
CLIPO (Cui et al., 2026)	78.08	48.33	38.92	24.03	27.25	42.68	33.20	55.56	76.94	47.22
MGS (Cai et al., 2026)	78.10	51.25	39.38	27.04	28.53	45.40	34.18	58.35	80.40	49.18
MT-GRPO (Ramesh et al., 2026)	78.53	52.70	40.16	27.41	28.34	44.06	34.52	58.67	79.93	49.37
CGPO (Liang et al., 2026)	79.06	52.88	40.54	27.75	28.23	43.85	34.69	58.90	79.25	49.46
MCPO-GRPO	81.18	54.20	43.22	28.56	30.72	48.68	34.91	**60.19**	**83.40**	51.67
MCPO-DAPO	**83.41**	**56.24**	**44.63**	**28.95**	**32.57**	**49.22**	**35.75**	59.89	82.63	**52.58**
Qwen3-8B										
Base Model (Team, 2025)	76.25	39.89	30.83	17.29	21.68	42.07	28.46	47.35	64.27	40.90
GRPO (Shao et al., 2024)	80.13	51.88	43.67	27.73	30.41	48.44	34.91	57.32	80.12	50.51
DAPO (Yu et al., 2025)	82.97	53.46	45.20	28.04	31.73	50.11	36.05	56.42	81.15	51.68
CLIPO (Cui et al., 2026)	79.38	51.56	43.44	28.46	30.64	49.83	34.32	57.69	78.86	50.46
MGS (Cai et al., 2026)	80.22	54.17	45.32	29.08	32.63	50.25	35.24	58.82	80.23	51.77
MT-GRPO (Ramesh et al., 2026)	80.94	54.05	44.67	30.33	32.18	49.89	35.71	59.40	81.63	52.09
CGPO (Liang et al., 2026)	81.77	54.80	46.15	31.07	32.73	50.56	36.29	60.33	79.63	52.59
MCPO-GRPO	82.94	57.87	47.43	32.56	34.27	50.76	36.88	**61.46**	84.23	54.27
MCPO-DAPO	**84.21**	**59.79**	**51.39**	**33.70**	**37.58**	**52.36**	**37.95**	61.02	**84.63**	**55.85**

Backbone Models. We adopt Qwen3 series (Team, 2025) as backbone models, specifically Qwen3-4B and Qwen3-8B. To examine whether the experimental conclusions generalize across different model families, we additionally evaluate MCPO on DeepSeek-R1-Distill-Qwen-7B and Llama3.1-8B-Instruct. Unless otherwise stated, algorithmic analyses are conducted on Qwen3-8B.

Baseline Methods. We compare MCPO against prompt-local and multi-domain RL baselines. GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and CLIPO (Cui et al., 2026) respectively use group-relative advantages, improved clipping and sampling, and contrastive supervision within each prompt. Among multi-domain methods, MGS (Cai et al., 2026) isolates learning directions through pooled gradient manipulation, MT-GRPO (Ramesh et al., 2026) balances learning progress through adaptive sampling and domain weighting, and CGPO (Liang et al., 2026) introduces second-order correction terms through randomized sequential domain updates to mitigate cross-domain interference. These baselines cover prompt-local optimization, direction isolation, progress balancing, and interaction correction. Detailed descriptions and comparisons are provided in Appendix C.2.

Tasks, Datasets, and Benchmarks. Following recent studies on general-purpose and multi-domain reasoning models (Wei et al., 2026; Bikkasani, 2025; Cai et al., 2026; Ramesh et al., 2026), we study joint RL post-training across five domains: mathematical reasoning (Math), code generation (Code), scientific question answering (Science), tool use (ToolUse), and safety alignment (Safety). These domains cover core LRM capabilities and expose a shared policy to heterogeneous forms. For training, we use 20,000 prompts drawn from five domain-specific datasets, and the per-domain sources are listed in Appendix C.1. We evaluate Math on AMC23 (Math-AI Team, 2024), AIME24 (HuggingFaceH4 Team, 2024), and AIME25 (OpenCompass Team, 2025); Code on LiveCodeBench v5 and v6 (Jain et al., 2024); Science on GPQA-Diamond (Rein et al., 2023) and SciBench (Wang et al., 2024); ToolUse on a ToolAIpaca (Tang et al., 2023) test; and Safety on Beaver (Ji et al., 2023) examples.

Evaluation Metrics. Following standard RLVR evaluation practice (Shao et al., 2024; Guo et al., 2025a; Yu et al., 2025), we sample 32 responses per problem in the zero-shot setting. The accuracy of each problem is the fraction of correct responses among its 32 samples, and we report the average of these per-problem accuracies across all problems as *avg@32*. For domains with multiple benchmarks, the domain score is the average of the *avg@32* scores across all the benchmarks. All methods use the same output formats and verifier rules, as detailed in Appendices C.4 and C.5.

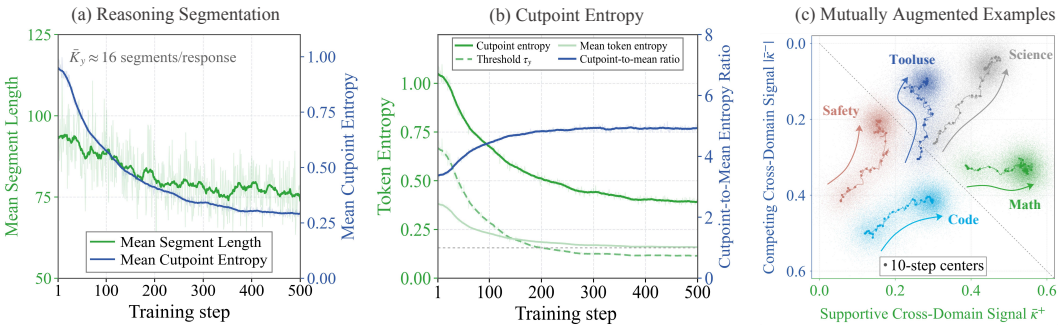

Figure 3: Reasoning segments and mutually augmented examples. (a) Mean segment length and mean cutpoint entropy. (b) Cutpoint entropy. Token entropy at the selected cutpoints stays far above the response-relative threshold and the mean token entropy. (c) Five-domain trajectories of cross-domain signals with ten-step centers. The diagonal denotes equal supportive and competing magnitudes.

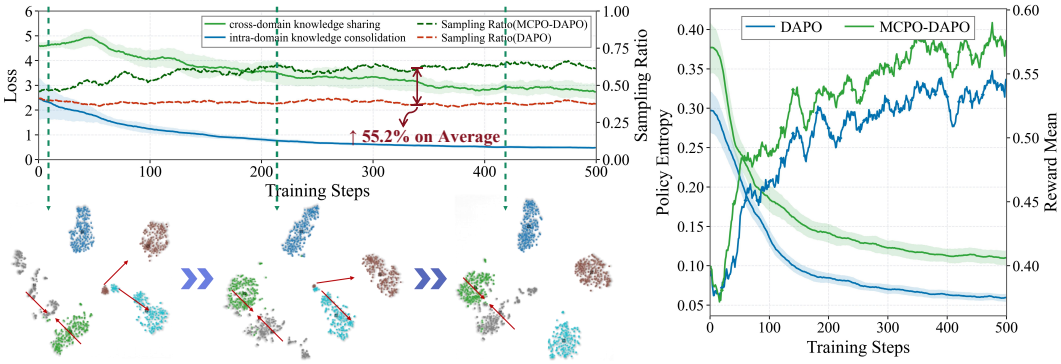

Figure 4: Knowledge sharing and policy exploration. Left: cross-domain knowledge sharing and intra-domain knowledge consolidation losses, sampling ratios, and t-SNE snapshots. Right: policy entropy and mean reward during the whole training process.

Implementation Details. All methods are implemented in the VERL framework (Sheng et al., 2025), and all experiments are run on a single node with 8 H200 GPUs. Full hyperparameters, prompt templates, reward rules, and complexity analysis are provided in Appendices C.3, C.4, and D.8.

4.2 Main Results

In this part, we present the experimental results and analyses to highlight the effectiveness of MCPO.

MCPO improves performance by utilizing positive and negative cross-domain interactions. Table 1 shows that contrastive supervision of signed interactions consistently improves both base algorithms. For example, on Qwen3-4B, MCPO-GRPO improves GRPO from 47.95 to 51.67, and MCPO-DAPO improves DAPO from 49.24 to 52.58; on Qwen3-8B, the corresponding gains are from 50.51 to 54.27 and from 51.68 to 55.85. The gains cover all five domains, ranging from 2.08 to 5.76 points. Panel (a) in Table 2 further shows that MCPO outperforms single-domain training in most cases, *e.g.*, Math, Code, and Science. These empirical results indicate that positive and negative cross-domain interactions can provide useful for knowledge sharing in multi-domain post-training.

MCPO enables knowledge sharing beyond interference alleviation. The baseline comparison in Table 1 further separates knowledge sharing from interference alleviation. DAPO strengthens prompt-local optimization, while CLIPO applies a learnable contrastive head only within each prompt group. Such coarse representation supervision cannot resolve reasoning structures or exploit relations across prompts and domains, leaving CLIPO 5.39 points behind MCPO-DAPO on Qwen3-8B. MGS and MT-GRPO directly target multi-domain training through gradient surgery and domain reweighting, but their averages of 51.77 and 52.09 still trail MCPO-DAPO, as does the strongest baseline CGPO by 3.26 points. Moreover, Panel (d) confirms the gains across backbones. These results suggest that indiscriminately mitigating cross-domain interactions is insufficient. MCPO instead gains by excavating and exploiting positive reasoning structures as supportive knowledge and treating negative reasoning structures as competing knowledge rather than mitigating them.

Table 2: Ablation studies on MCPO. M-*Avg.*, C-*Avg.*, and S-*Avg.* denote the averages of Math, Code, and Science, respectively. Panel (a) ablates the contrastive losses and compares mixed-domain DAPO with single-domain DAPO; panel (b) varies the maximum number of reasoning segments; panel (c) varies the relation retention ratio Top-q_R; and panel (d) tests backbone generalization.

(a) Ablation study on contrastive losses.

Variant	M-*Avg.*	C-*Avg.*	S-*Avg.*	Tool	Safety	*Avg.*
DAPO (Mixed)	60.54	29.89	43.08	56.42	81.15	51.68
DAPO (Single)	64.86	34.96	43.85	**62.00**	**85.28**	55.50
Intra-domain only	63.81	32.15	43.74	58.74	81.87	53.76
Cross-domain only	63.88	32.54	44.45	57.18	82.50	53.92
MCPO-DAPO	**65.14**	**35.64**	**45.16**	61.02	84.63	**55.85**

(b) Ablation on the maximum number of segments.

$K_{\max}$	M-*Avg.*	C-*Avg.*	S-*Avg.*	*Avg.*
8	64.25	34.81	44.09	54.08
12	64.77	35.34	45.02	55.37
16	**65.14**	**35.64**	**45.16**	**55.85**
24	65.10	35.43	45.09	55.65
32	65.02	35.36	44.97	55.46
64	64.68	34.51	44.63	55.02

(c) Ablation on the relation-retention ratio.

Top-q_R	M-*Avg.*	C-*Avg.*	S-*Avg.*	Tool	Safety	*Avg.*
10%	64.69	35.05	44.87	60.75	84.60	55.47
20%	**65.14**	35.64	45.16	**61.02**	**84.63**	**55.85**
30%	65.03	**35.82**	**45.45**	60.50	84.38	55.83

(d) Analysis of backbone generalization.

Backbone	Base	GRPO	MGS	MCPO
DS-7B	33.00	40.18	42.75	**44.32**
Llama-8B	22.41	31.93	31.68	**34.85**

4.3 Algorithmic Analyses

In this part, we conduct in-depth investigations on our proposed MCPO to analyze its effectiveness. In addition, Appendix D further analyzes the phase handover of cross-domain interactions, the geometry of the learned reasoning space, the application of cross-batch source bank, and the case study.

Reasoning segmentation captures underlying reasoning structures. Figure 3(a) shows that entropy-based segmentation tracks the underlying reasoning structures as training proceeds. Figure 3(b) further validates entropy as the boundary cue, where the selected cutpoints consistently sit on high-entropy forks far above the threshold and the mean token entropy. The cutpoint-to-mean entropy ratio correspondingly rises from 3.5× to 4.9× as training sharpens the policy. The mean cutpoint entropy declines together with the policy entropy in Figure 4, while the mean segment length shortens in a coordinated manner. Because the segments preserve trajectory context and recover the response gradient when aggregated, they expose interactions hidden by prompt-level aggregation. Panel (b) further supports a moderate granularity for constructing mutually augmented examples.

Positive and negative segment pairs strengthen knowledge sharing. Figure 3(c) shows how the gradient variation test organizes positive and negative segment pairs into mutually augmented examples. Each domain obtains higher supportive signals to varying degrees during training, while competing signals are discriminably separated. Beyond the pair level, the signed interactions keep handing over among domain pairs throughout training, and the learned reasoning space simultaneously evolves with compatible domains approaching and competing domains separating, as detailed in Appendices D.1 and D.2. Furthermore, Panel (c) peaks at 55.85 when Top-q_R is 20% and remains at 55.83 with 30%, which shows that MCPO can stably utilize interactions for knowledge sharing.

MCPO forms a harmonious reasoning space while enhancing policy exploration. Figure 4 and Appendix D.2 jointly show how knowledge sharing appears in the geometry of the reasoning space. As the two contrastive losses decrease, Math and Science move closer, while Safety and ToolUse become separated, exhibiting cross-domain compatibility and intra-domain discriminability. Panel (a) confirms their complementarity, as intra-domain consolidation and cross-domain sharing alone reach 53.76 and 53.92, whereas their combination reaches 55.85. Regarding policy exploration, contrastive learning may intuitively reduce exploration by pulling compatible structures together. In practice, MCPO-DAPO instead obtains a 55.2% higher average sampling ratio than DAPO, together with higher policy entropy benefiting from extensive competition of negative pairs throughout training.

5 Conclusion

We propose MCPO, a multi-domain RL framework that exploits interactions among the underlying reasoning structures of rollouts generated by LRMs. We argue that cross-domain interactions are not exclusively harmful and transform their positive and negative relations into mutually augmented examples carrying supportive and competing knowledge. Therefore, MCPO uses these supportive and competing signals for cross-domain knowledge sharing and intra-domain knowledge consolidation. Accordingly, MCPO makes reasoning structures compatible across domains and discriminable within each domain, thereby forming a harmonious reasoning space. Extensive experiments across Math, Code, Science, ToolUse, and Safety show that MCPO consistently improves GRPO and DAPO, alleviates mixed-domain degradation, and outperforms single-domain training in most cases.

AI USE STATEMENT

In this work, large language models are the primary subject of study, and all experiments involve multi-domain RL post-training of LRMs with rule-based verifiers, including answer verification, code execution, structured tool-call matching, and safety preference labels. We did not use generative AI tools to propose or refine research hypotheses, develop the conceptual framework, design the methodology or experiments, implement the proposed method, or analyze or interpret the resulting evaluation scores. The required-disclosure tasks concerning theoretical models, mathematical claims and proofs, and translation are not applicable to this work. Additionally, we used generative AI tools for language editing to improve the grammar, clarity, concision, and readability of the manuscript.

We reviewed all AI-assisted work. Every AI-assisted language edit was examined by the authors and retained only after confirming that it preserved the intended technical meaning. For the evaluation metrics, we followed the protocol presented in Appendix C.5 and applied the same verifier procedure consistently to all compared methods within each benchmark. The authors verified the evaluation implementation and score aggregation and independently analyzed and interpreted the resulting scores. All research ideas, technical decisions, experimental conclusions, and scientific claims were developed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We document the information needed to independently re-implement MCPO and reproduce its experimental results throughout the paper and appendices.

For MCPO, the efforts that have been made to ensure reproducibility are as follows.

- Section 3 presents the overall framework and formalizes the multi-domain objective, reasoning segmentation, the gradient variation test, and the two contrastive objectives.
- Appendix B derives the signed decomposition of cross-prompt interactions, the entropy-based filtering rule, and the output-gradient proxy of the interaction score.
- Appendix C.3 specifies the implementation settings, including the rollout counts, sequence limits, learning rate, MCPO loss coefficients, entropy quantile, and relation-retention ratio.
- Appendix C.4 documents the unified prompt templates and the verifier rules of the five reasoning domains.

For the experimental results, the efforts that have been made to ensure reproducibility are as follows.

- Section 4.1 describes the experimental setup, including the baseline methods, backbone models, benchmarks, and evaluation protocols.
- Appendix C.2 provides the implementation details of each baseline method.
- Appendix C.1 presents the descriptions of the adopted datasets and the per-domain allocation for training and evaluation.
- Appendix C.5 provides the definitions and calculation protocols of the evaluation metrics in the experiments.
- Table 2 specifies the configurations of the ablated MCPO variants in the algorithmic analyses.

REFERENCES

Zhengyang Ai, Zikang Shan, Xiaodong Ai, Jingxian Tang, Hangkai Hu, and Pinyan Lu. SHAPE: Stage-aware hierarchical advantage via potential estimation for LLM reasoning. In *ACL*, pp. 24288–24305, 2026.

Dileesh Chandra Bikkasani. Navigating artificial general intelligence (AGI): societal implications, ethical considerations, and governance strategies. *AI Ethics*, 5(3):2021–2036, 2025.

Min Cai, Yu Liang, Longzheng Wang, Yan Wang, Yueyang Zhang, Long Xia, Zhiyuan Sun, Xi Ye, and Daiting Shi. Advancing general-purpose reasoning models with modular gradient surgery. *CoRR*, abs/2602.02301, 2026.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607, 2020.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pp. 793–802, 2018.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, pp. 4299–4307, 2017.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617, 2025.

Sijia Cui, Pengyu Cheng, Jiajun Song, Yongbo Gai, Guojun Zhang, Zhechao Yu, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. CLIPO: contrastive learning in policy optimization generalizes RLVR. *CoRR*, abs/2603.10101, 2026.

Lishui Fan, Yu Zhang, Mouxiang Chen, and Zhongxin Liu. Posterior-grpo: Rewarding reasoning processes in code generation. *CoRR*, abs/2508.05170, 2025.

Kehua Feng, Keyan Ding, Weijie Wang, Xiang Zhuang, Zeyuan Wang, Ming Qin, Yu Zhao, Jianhua Yao, Qiang Zhang, and Huajun Chen. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *CoRR*, abs/2406.09098, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, Tao Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638, 2025a.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in RL for large language models. *CoRR*, abs/2505.23564, 2025b.

Feng He, Zijun Chen, Xinnian Liang, Tingting Ma, Yunqi Qiu, Shuangzhi Wu, and Junchi Yan. Protoreasoning: Prototypes as the foundation for generalizable reasoning in llms. *CoRR*, abs/2506.15211, 2025.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9726–9735, 2020.

HuggingFaceH4 Team. aime_2024. Hugging Face dataset, 2024.

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

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. *CoRR*, abs/2307.04657, 2023.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pp. 7482–7491, 2018.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

Xize Liang, Lin Yang, Jie Wang, Rui Liu, Yang Lu, Jinliang Zeng, Hanzhu Chen, Dong Li, and Jianye Hao. Boosting multi-domain reasoning of LLMs via curvature-guided policy optimization. In *ICLR*, 2026.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, pp. 18878–18890, 2021.

Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. GitHub repository, 2025.

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

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling. *CoRR*, abs/2504.02495, 2025.

Math-AI Team. Amc23. Hugging Face dataset, 2024.

nlile. Numinamath-1.5-rl-verifiable, 2025.

OpenCompass Team. Aime2025. Hugging Face dataset, 2025.

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 *NeurIPS*, 2022.

Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *KDD*, pp. 239–247, 2013.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. *CoRR*, abs/1905.06922, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

Shyam Sundhar Ramesh, Xiaotong Ji, Matthieu Zimmer, Sangwoong Yoon, Zhiyong Wang, Haitham Bou-Ammar, Aurélien Lucchi, and Ilija Bogunovic. Multi-task GRPO: reliable LLM reasoning across tasks. *CoRR*, abs/2602.05547, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023.

Xuankun Rong, Wenke Huang, Tingfeng Wang, Daiguo Zhou, Bo Du, and Mang Ye. Safe-grpo: Self-rewarded multimodal safety alignment via rule-governed policy optimization. *CoRR*, abs/2511.12982, 2025.

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

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. In *EuroSys*, pp. 1279–1297, 2025.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *CoRR*, abs/2306.05301, 2023.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. In *ICML*, pp. 50622–50649, 2024.

Yikun Wang, Yibin Wang, Dianyi Wang, Zimian Peng, Qipeng Guo, Dacheng Tao, and Jiaqi Wang. Geometryzero: Improving geometry solving for LLM with group contrastive policy optimization. *CoRR*, abs/2506.07160, 2025.

Qikai Wei, Huansheng Ning, Feifei Shi, Tao Zhu, Jianguo Ding, Abdelouahid Derhab, and Mahmoud Daneshmand. Agi-enabled solutions for service explosion in iox: A cyber-physical-social-thinking perspective. *ACM Comput. Surv.*, 58(8):208:1–208:32, 2026.

Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. Imbalanced gradients in RL post-training of multi-task llms. In *EACL (Findings)*, pp. 3137–3150, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022.

Zepeng Zhai, Meilin Chen, Jiaxuan Zhao, Junlang Qian, Lei Shen, and Yuan Lu. Rewards as labels: Revisiting RLVR from a classification perspective. *CoRR*, abs/2602.05630, 2026.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *CoRR*, abs/2507.18071, 2025.

Kejian Zhu, Zhuoran Jin, Shangqing Tu, Hongbang Yuan, Yushi Bai, Kang Liu, Juanzi Li, and Jun Zhao. SFT conflicts, RL coexists: A theoretical and empirical analysis of multi-task learning paradigms for LLMs. *CoRR*, abs/2608.03573, 2026.

A Related Work

A.1 GRPO and Prompt-Local Policy Optimization

Reinforcement learning with verifiable rewards has become a common route for improving LRM capabilities after supervised fine-tuning, since verifier- or rule-based rewards provide scalable supervision (Christiano et al., 2017; Ouyang et al., 2022; Zelikman et al., 2022; Liu et al., 2026). GRPO (Shao et al., 2024; Guo et al., 2025a; Zheng et al., 2025) estimates relative advantages within the rollout group of each prompt and efficiently optimizes a shared policy without a separate value model. Prompt-local variants strengthen this regime in different ways, for example the asymmetric clipping, dynamic sampling, and token-level aggregation introduced by DAPO (Yu et al., 2025) and the learnable contrastive head with which CLIPO (Cui et al., 2026) supervises each rollout group in representation space. These methods organize supervision strictly within each prompt, so their updates retain the prompt-local descent component but cannot expose which reasoning structures across prompts or domains support or oppose one another. MCPO keeps the GRPO reward and advantage computation unchanged and instead resolves these cross-prompt interactions before aggregation.

A.2 Multi-domain Reinforcement Learning

When heterogeneous domains are mixed, the update contains a prompt-local component and a cross-prompt interaction component, as made explicit in Eq. (3). Existing methods handle the interaction component through two broad strategies. One strategy suppresses or projects away cross-domain interactions to isolate learning directions, including gradient surgery and conflict-averse updates (Yu et al., 2020; Liu et al., 2021; Cai et al., 2026). The other retains the aggregate interaction but adjusts adaptive domain weights ω_d, distinct from the fixed mixture weights α_d, to balance learning progress through uncertainty weighting, gradient normalization, or task-aware sampling (Kendall et al., 2018; Chen et al., 2018; Ramesh et al., 2026). Curvature-guided optimization further adds second-order corrections to this domain-level coordination (Liang et al., 2026), while comparative analyses of SFT and RL report distinct gradient-interaction regimes that motivate measuring the RL geometry directly (Zhu et al., 2026). These approaches control interference or imbalance after local gradients have been pooled into domain-level aggregates, so they do not preserve which reasoning segments support or compete with one another. MCPO instead exposes signed segment-level relations before aggregation, using positive relations for cross-domain knowledge sharing and intra-domain knowledge consolidation while retaining negative relations as contrastive competitors.

A.3 Contrastive Learning

Contrastive learning learns informative representations by increasing the similarity of positive pairs and decreasing that of negative pairs, typically through InfoNCE-style objectives (van den Oord et al., 2018; Chen et al., 2020; He et al., 2020; Radford et al., 2021; Khosla et al., 2020; Poole et al., 2019). This principle matches LRM reasoning because different rollouts can contain mutually supportive reasoning structures despite diverse complete chains of reasoning (He et al., 2025). Recent studies on LLM reasoning and RLVR have accordingly introduced process rewards or auxiliary contrastive objectives that provide supervision beyond sparse outcome rewards (Wang et al., 2025; Liu et al., 2025; Cui et al., 2026; Zhai et al., 2026). However, existing contrastive integrations remain prompt-local, strengthening within-prompt supervision without identifying reusable reasoning structures from other prompts or domains. MCPO extends contrastive learning to signed relations among reasoning segments in multi-domain RL, organizing them as mutually augmented examples for cross-domain knowledge sharing and intra-domain knowledge consolidation, thereby forming a harmonious reasoning space with cross-domain compatibility and intra-domain discriminability.

B MATHEMATICAL DERIVATIONS

This appendix gives the proof chain used by MCPO. We first write the sampled GRPO update in the notation of Section 2, prove its exact regrouping over contiguous segments, and identify the signed segment interactions hidden by prompt aggregation (Appendix B.1). We then justify entropy as a candidate-cutpoint statistic (Appendix B.2), derive the local source-to-target transfer criterion (Appendix B.3), and establish when the compact interaction score preserves the sign of the ideal full-gradient interaction (Appendix B.4). Finally, we derive the score-level direction of the two multi-positive contrastive losses (Appendix B.5), connect their pairwise coefficients to the total update in Eq. (12), and derive conditional batch-level and stability results (Appendix B.6). All losses below are sampled GRPO surrogates.

B.1 MIXED-DOMAIN GRPO AND THE UNRESOLVED LOCAL ASSIGNMENT

Fix a sampled batch $\mathcal{B}$ of prompts. For prompt p, let $T_{p,g} = |y_{p,g}|$ and $N_p = \sum_{g=1}^{G} T_{p,g}$; hence $N_{\mathcal{B}} = \sum_{p\in\mathcal{B}} N_p$. During one policy update, the rollouts, rewards, advantages, and old-policy probabilities are held fixed. We consider nonempty responses and $N_p > 0$. The domain weights α_d in Eq. (1) govern prompt sampling, whereas the realized token weights below govern batch aggregation. Define the prompt-normalized token contribution by

$$q_{p,g,t}(\boldsymbol{\theta}) = -\frac{1}{N_p}\min(\rho_{p,g,t}A_{p,g}, \mathrm{clip}(\rho_{p,g,t}, 1-\varepsilon, 1+\varepsilon)A_{p,g})\,. \tag{13}$$

Set $\mathcal{L}_p = \sum_{g=1}^{G}\sum_{t=1}^{T_{p,g}} q_{p,g,t}$. With $w_p = N_p/N_{\mathcal{B}}$, the token-mean objective in Eq. (2) is exactly $\mathcal{L}_{\mathrm{GRPO}} = \sum_{p\in\mathcal{B}} w_p\mathcal{L}_p$. Let $\boldsymbol{g}_p = \nabla_{\boldsymbol{\theta}}\mathcal{L}_p$ and $\boldsymbol{g}_{\mathcal{B}} = \sum_{p\in\mathcal{B}} w_p\boldsymbol{g}_p$. The sampled update is

$$\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta\boldsymbol{g}_{\mathcal{B}}. \tag{14}$$

This fixes the normalization left implicit in Eq. (3). Its algebra also holds for any fixed normalized nonnegative prompt weights. The virtual steps below are Euclidean gradient steps. At clipping kinks, we use a consistent computational derivative for additive identities; the smoothness-based results apply on neighborhoods where the losses have Lipschitz gradients, for example when no active clipping branch changes.

Proposition 1 (Target-prompt effect of the mixed GRPO update)**.** *Assume that $\mathcal{L}_p$ has an L_p-Lipschitz gradient on the line segment between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^+$. Then*

$$\begin{aligned}\mathcal{L}_p(\boldsymbol{\theta}^+) - \mathcal{L}_p(\boldsymbol{\theta}) &= -\eta\boldsymbol{g}_p^\top\boldsymbol{g}_{\mathcal{B}} + R_p \\ &= -\eta w_p\|\boldsymbol{g}_p\|_2^2 - \eta\sum_{p'\in\mathcal{B}\setminus\{p\}} w_{p'}\boldsymbol{g}_p^\top\boldsymbol{g}_{p'} + R_p, \qquad |R_p| \le \frac{L_p\eta^2}{2}\|\boldsymbol{g}_{\mathcal{B}}\|_2^2.\end{aligned} \tag{15}$$

Proof. The descent lemma applied to the displacement $-\eta\boldsymbol{g}_{\mathcal{B}}$ gives

$$\mathcal{L}_p(\boldsymbol{\theta} - \eta\boldsymbol{g}_{\mathcal{B}}) = \mathcal{L}_p(\boldsymbol{\theta}) - \eta\boldsymbol{g}_p^\top\boldsymbol{g}_{\mathcal{B}} + R_p, \qquad |R_p| \le \frac{L_p\eta^2}{2}\|\boldsymbol{g}_{\mathcal{B}}\|_2^2. \tag{16}$$

Substituting the weighted sum defining $\boldsymbol{g}_{\mathcal{B}}$ and separating the term $p' = p$ yields Eq. (15). □

The first term in Eq. (15) is the descent contribution from the target prompt itself. The second is the complete first-order contribution of all other prompts, giving Eq. (3) with an explicit remainder. Using $z = [z]_+ - [-z]_+$ for every inner product gives Eq. (4) exactly. In particular, centered advantages alone do not imply that the cross-prompt sum vanishes. Thus shared parameters transmit cross-prompt effects through the update. The unresolved object is the assignment of this sum to the local reasoning decisions that generated the two prompt gradients; GRPO contains no separate variable for that assignment.

For a realized partition of rollout $y_{p,g}$, let $\mathcal{I}_{p,g,k} = \{b_{p,g,k-1}+1, \ldots, b_{p,g,k}\}$ be the token indices of segment $s_{p,g}^k$, where $K_{p,g} = K_{y_{p,g}}$ and the boundaries are those of Eq. (6) for that rollout. Define

$$\ell_{p,g,k} = \sum_{t\in\mathcal{I}_{p,g,k}} q_{p,g,t}, \qquad \boldsymbol{g}_{p,g,k} = \nabla_{\boldsymbol{\theta}}\ell_{p,g,k}. \tag{17}$$

Proposition 2 (Exact segment decomposition)**.** *If the segment index sets are pairwise disjoint and cover* $\{1, \dots, T_{p,g}\}$ *for every rollout, then*

$$\mathcal{L}_p = \sum_{g=1}^{G} \sum_{k=1}^{K_{p,g}} \ell_{p,g,k}, \qquad \boldsymbol{g}_p = \sum_{g=1}^{G} \sum_{k=1}^{K_{p,g}} \boldsymbol{g}_{p,g,k}. \tag{18}$$

At a ratio-clipping boundary, the equality holds for the same per-token computational derivative on both sides.

Proof. Disjoint coverage partitions the finite token sum, so regrouping gives

$$\sum_{g=1}^{G} \sum_{t=1}^{T_{p,g}} q_{p,g,t} = \sum_{g=1}^{G} \sum_{k=1}^{K_{p,g}} \sum_{t \in \mathcal{I}_{p,g,k}} q_{p,g,t}. \tag{19}$$

Differentiation is linear over this finite sum wherever the clipped loss is differentiable. At a boundary, selecting the same per-token derivative on both sides preserves the equality. The construction introduces no segment reward, segment advantage, or token-credit redistribution. □

For brevity, write $u = (p, g, k)$ and $v = (p', g', k')$ for two segments and $\boldsymbol{g}_u = \boldsymbol{g}_{p,g,k}$. Let $\mathcal{S}(p)$ collect the segments of all G rollouts, indexed by their parent rollout and boundaries, so repeated text spans remain distinct. With $\boldsymbol{g}_t = \nabla_{\boldsymbol{\theta}} q_{p,g,t}$ within a fixed rollout, $\boldsymbol{g}_u = \sum_{t \in \mathcal{I}(u)} \boldsymbol{g}_t$, which also gives the token-pair expansion in Eq. (8). Every token retains the full prompt and preceding response prefix. The prompt interaction is therefore

$$\boldsymbol{g}_p^\top \boldsymbol{g}_{p'} = \sum_{g,k} \sum_{g',k'} \boldsymbol{g}_{p,g,k}^\top \boldsymbol{g}_{p',g',k'} = \sum_{u \in \mathcal{S}(p)} \sum_{v \in \mathcal{S}(p')} \boldsymbol{g}_u^\top \boldsymbol{g}_v. \tag{20}$$

Define the ideal ordered interaction $\kappa_{u \leftarrow v} = \boldsymbol{g}_u^\top \boldsymbol{g}_v$. Although this inner product is symmetric at a fixed parameter value, the source-to-target effect used by MCPO is ordered because v supplies the virtual update and u supplies the loss being evaluated.

Let $\mathcal{R}_+$, $\mathcal{R}_-$, and $\mathcal{R}_0$ be the positive, negative, and zero terms in Eq. (20). Then

$$\boldsymbol{g}_p^\top \boldsymbol{g}_{p'} = \sum_{(u,v) \in \mathcal{R}_+} |\kappa_{u \leftarrow v}| - \sum_{(u,v) \in \mathcal{R}_-} |\kappa_{u \leftarrow v}|. \tag{21}$$

This identity explains how a small prompt-level interaction can contain large opposed local terms. It establishes a possible cancellation pattern; its frequency is an empirical quantity.

Proposition 3 (Non-identifiability after prompt aggregation)**.** *Let* m *and* n *be the numbers of segments in two prompts and let* $\mathbf{K} \in \mathbb{R}^{m \times n}$ *contain the entries* $K_{k,k'} = \kappa_{u_k \leftarrow v_{k'}}$*. If* $mn > 1$*, the scalar aggregation map* $\Phi(\mathbf{K}) = \mathbf{1}^\top \mathbf{K} \mathbf{1}$ *is non-injective. Hence the prompt-level interaction cannot identify the location or sign pattern of its segment interactions.*

Proof. The linear map $\Phi : \mathbb{R}^{m \times n} \to \mathbb{R}$ has a null space of dimension at least $mn - 1$. Choose two distinct entries and put c in one and $-c$ in the other, with all remaining entries zero. The resulting nonzero matrix $\mathbf{Q}$ satisfies $\Phi(\mathbf{Q}) = 0$, so $\Phi(\mathbf{K}) = \Phi(\mathbf{K} + \mathbf{Q})$ while the local values and their signs can differ. The ambiguity also occurs for realizable gradient interactions with the same aggregate gradient. In one parameter dimension, two source segments with gradients $(c, -c)$ and $(0, 0)$ both sum to zero, but against a target segment of gradient 1 they yield different signed pairs. If only the target has multiple segments, exchange the two roles. Thus the ambiguity is not an artifact of allowing arbitrary matrices. □

For a diagnostic of the hidden signed mass, define

$$M_{p,p'}^{+} = \sum_{u,v} [\kappa_{u \leftarrow v}]_+, \qquad M_{p,p'}^{-} = \sum_{u,v} [-\kappa_{u \leftarrow v}]_+,$$

$$I_{p,p'} = M_{p,p'}^{+} - M_{p,p'}^{-}, \qquad C_{p,p'} = 1 - \frac{|I_{p,p'}|}{M_{p,p'}^{+} + M_{p,p'}^{-} + \epsilon}. \tag{22}$$

The triangle inequality gives $0 \le C_{p,p'} \le 1$. A value near one is evidence of cancellation only when the total signed mass is appreciable; if both masses vanish, the stabilized value carries no interaction evidence. The corresponding prompt-level masses in Eq. (4) satisfy

$$M_p^{\pm} \le \sum_{p' \ne p} w_{p'} M_{p,p'}^{\pm}, \qquad M_p^{+} - M_p^{-} = \sum_{p' \ne p} w_{p'} (M_{p,p'}^{+} - M_{p,p'}^{-}). \tag{23}$$

Indeed, $[\sum_j z_j]_+ \le \sum_j [z_j]_+$, applied separately to each prompt pair and to its negative, proves the inequalities; subtraction uses Eq. (20). Segment-level resolution can expose opposed mass already canceled inside a prompt pair while preserving the same aggregate first-order effect.

B.2 Entropy-defined reasoning segmentation

For response $y_{p,g} = (y_1, \dots, y_T)$, define the next-token distribution under the rollout policy and its entropy (using natural logarithms) by

$$p_t(a) = \pi_{\boldsymbol{\theta}_{\mathrm{old}}}(a \mid p, y_{<t}), \qquad H_t = -\sum_{a \in \mathcal{V}} p_t(a) \log p_t(a), \qquad t = 1, \dots, T-1. \tag{24}$$

For the response-relative threshold used in Section 3.1, $\tau_y = Q_{q_H}(\{H_t\}_{t=1}^{T-1})$ and

$$\mathcal{C}_{\tau_y}(y) = \{t \in \{1, \dots, T-1\} : H_t \ge \tau_y\}. \tag{25}$$

For $T = 1$, no quantile is needed and the response forms one segment. For $T \ge 2$, Q_{q_H} denotes linear interpolation between consecutive order statistics: if $a = 1 + (T-2)q_H$, $j = \lfloor a \rfloor$, and $\delta = a - j$, the threshold is $(1-\delta)H_{(j)} + \delta H_{(\min(j+1, T-1))}$. Ties at the threshold are included. The candidates are ordered as $c_1 < \cdots < c_m$. If $m \le K_{\max} - 1$, all candidates are retained before the minimum-span filter. Otherwise, the ordered downsample is

$$\widetilde{c}_j = c_{\lceil jm/K_{\max} \rceil}, \qquad j = 1, \dots, K_{\max} - 1. \tag{26}$$

Here $K_{\max} \ge 1$ is the segment budget and $m_{\min} \ge 1$ is the minimum span. The filter scans these candidates from left to right. A candidate is discarded if either endpoint span is shorter than $m_{\min}$; it is appended when its distance from the preceding retained cut is at least $m_{\min}$; when two candidates conflict, the one with larger H_t replaces the previous one only if the resulting preceding span is valid. Equal-entropy conflicts keep the earlier cut. If no cut survives, the response remains one segment; when $T < m_{\min}$ this sole segment is shorter than the minimum span. The final retained cuts are $b_1 < \cdots < b_{K_y - 1}$, with $b_0 = 0$ and $b_{K_y} = T$, and define

$$s^k = y_{b_{k-1}+1:b_k}, \qquad \mathcal{I}(s^k) = \{b_{k-1} + 1, \dots, b_k\}, \qquad y = s^1 \oplus \cdots \oplus s^{K_y}. \tag{27}$$

Following Eq. (5), the mathematical partition uses the rollout-policy entropy and is fixed during differentiation. Strictly increasing boundaries ensure disjoint coverage, and the filter gives $K_y \le K_{\max}$. Therefore Eq. (18) applies to the realized partition for every response.

Proposition 4 (Entropy and categorical score dispersion)**.** *Let* $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$ *and* $Y \sim \boldsymbol{p}$*. Then*

$$\mathbb{E}\Big[\|\nabla_{\boldsymbol{z}} \log p(Y)\|_2^2\Big] = \sum_a p_a \|\boldsymbol{e}_a - \boldsymbol{p}\|_2^2 = 1 - \|\boldsymbol{p}\|_2^2 = \mathrm{tr}\big(\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top\big). \tag{28}$$

Proof. The softmax score is $\nabla_{\boldsymbol{z}} \log p(a) = \boldsymbol{e}_a - \boldsymbol{p}$ for each vocabulary entry a, so $\|\boldsymbol{e}_a - \boldsymbol{p}\|_2^2 = 1 - 2p_a + \|\boldsymbol{p}\|_2^2$. Taking expectation over Y gives $1 - 2\sum_a p_a^2 + \|\boldsymbol{p}\|_2^2 = 1 - \|\boldsymbol{p}\|_2^2$. The last equality is the trace of the categorical Fisher matrix with respect to the logits. □

Both $H(\boldsymbol{p})$ and $1 - \|\boldsymbol{p}\|_2^2$ are Schur-concave, so a more dispersed predictive distribution increases each statistic under majorization. For example, transferring probability mass δ from a coordinate with probability c_1 to one with $c_2 < c_1$, for $0 < \delta \le (c_1 - c_2)/2$, increases $1 - \|\boldsymbol{p}\|_2^2$ by $2\delta(c_1 - c_2 - \delta)$. The derivative of the corresponding entropy change is $\log((c_1 - \delta)/(c_2 + \delta)) \ge 0$. Such averaging operations establish the stated dispersion relation. Entropy and Fisher trace need not rank arbitrary distributions identically, and the expectation above samples a new token from the categorical distribution at a fixed prefix. It is not a lower bound on the gradient of the particular realized token or on a parameter-space gradient. MCPO uses entropy-based candidates following SHAPE (Ai et al., 2026) and controlled contiguous partitions following SPO (Guo et al., 2025b). The transfer test supplies the relation sign. The present construction adds no stage reward or segment advantage.

B.3 LOCAL SOURCE-TO-TARGET TRANSFER

For target segment u and source segment v, let ℓ_u be the target loss in Eq. (17), $\boldsymbol{g}_u = \nabla_{\boldsymbol{\theta}} \ell_u$, and $\boldsymbol{g}_v = \nabla_{\boldsymbol{\theta}} \ell_v$. A source descent step is the virtual update $\boldsymbol{\theta}_v = \boldsymbol{\theta} - \eta \boldsymbol{g}_v$.

Proposition 5 (First-order segment transfer)**.** *Fix $\eta > 0$ and evaluate both gradients at $\boldsymbol{\theta}$; the source vector is held fixed throughout the virtual step. Assume that ℓ_u has an L_u-Lipschitz gradient on the line segment from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_v$. Then*

$$\Delta_{u \leftarrow v}(\eta) = \ell_u(\boldsymbol{\theta} - \eta \boldsymbol{g}_v) - \ell_u(\boldsymbol{\theta}) = -\eta \kappa_{u \leftarrow v} + R_{u,v}, \qquad |R_{u,v}| \le \frac{L_u \eta^2}{2} \|\boldsymbol{g}_v\|_2^2, \tag{29}$$

where $\kappa_{u \leftarrow v} = \boldsymbol{g}_u^\top \boldsymbol{g}_v$. In particular, the sign predicted by $\kappa_{u \leftarrow v}$ is reliable under the local bound when

$$|\kappa_{u \leftarrow v}| > \frac{L_u \eta}{2} \|\boldsymbol{g}_v\|_2^2. \tag{30}$$

Proof. By the fundamental theorem of calculus,

$$\begin{aligned} \Delta_{u \leftarrow v}(\eta) &= -\eta \int_0^1 \nabla \ell_u(\boldsymbol{\theta} - t\eta \boldsymbol{g}_v)^\top \boldsymbol{g}_v \, dt \\ &= -\eta \boldsymbol{g}_u^\top \boldsymbol{g}_v - \eta \int_0^1 \left[\nabla \ell_u(\boldsymbol{\theta} - t\eta \boldsymbol{g}_v) - \nabla \ell_u(\boldsymbol{\theta})\right]^\top \boldsymbol{g}_v \, dt. \end{aligned} \tag{31}$$

The Lipschitz assumption and Cauchy–Schwarz bound the second term by $\eta \int_0^1 L_u t\eta \|\boldsymbol{g}_v\|_2^2 dt$, which equals the remainder bound in Eq. (29), after taking absolute values. The sign condition follows by requiring the first-order term to exceed that bound. A positive ideal interaction predicts a target-loss decrease for a sufficiently small source step, while a negative interaction predicts a local increase. □

B.4 COMPACT INTERACTION SCORES AND SIGN PRESERVATION

MCPO constructs relation sets from a compact gradient interaction, making segment-level supervision feasible without materializing a full model gradient for every segment. This subsection establishes its optimization meaning in three steps: the output-block interaction has an exact gradient factorization and a local loss-variation interpretation; residual approximation admits a probability-tail bound, while positive normalization preserves signs exactly; and an interaction-margin condition connects the compact score to full-parameter transfer. These results explain the construction and its sufficient conditions. The matched controls in Appendix D.4 separately test whether preserving the relation structure improves downstream learning.

Gradient factors and compact scores. For an independently parameterized output matrix $\boldsymbol{W}$, let $\boldsymbol{h}_t$ be its input and $\boldsymbol{r}_t = \boldsymbol{p}_t - \boldsymbol{e}_{y_t}$ the categorical residual. With the token loss defined in Eq. (13), set

$$\alpha_t = -\frac{\partial q_{p,g,t}}{\partial \log \pi_{\boldsymbol{\theta}}(y_{p,g,t} \mid p, y_{p,g,<t})}, \qquad \boldsymbol{G}_s = \nabla_{\boldsymbol{W}} \ell_s = \sum_{t \in \mathcal{I}(s)} \alpha_t \boldsymbol{r}_t \boldsymbol{h}_t^\top. \tag{32}$$

The coefficient includes the advantage, importance ratio, clipping mask, and prompt normalization. Bilinearity gives the exact output-block interaction

$$\langle \boldsymbol{G}_u, \boldsymbol{G}_v \rangle_F = \sum_{t \in \mathcal{I}(u)} \sum_{r \in \mathcal{I}(v)} \alpha_t \alpha_r (\boldsymbol{r}_t^\top \boldsymbol{r}_r)(\boldsymbol{h}_t^\top \boldsymbol{h}_r). \tag{33}$$

Thus the score combines the local prediction residual with the contextual hidden state, weighted by the policy-loss coefficient, rather than using hidden-state similarity alone. Let $\kappa^W_{u \leftarrow v} = \langle \boldsymbol{G}_u, \boldsymbol{G}_v \rangle_F$ and let $\Delta^W_{u \leftarrow v}(\eta)$ denote the change in the target loss after the virtual step $\boldsymbol{W} \mapsto \boldsymbol{W} - \eta \boldsymbol{G}_v$, with the remaining parameters fixed. Under local $L_{u,W}$-smoothness in this block, the argument of Proposition 5 gives

$$\left|\Delta^W_{u \leftarrow v}(\eta) + \eta \kappa^W_{u \leftarrow v}\right| \le \frac{L_{u,W} \eta^2}{2} \|\boldsymbol{G}_v\|_F^2. \tag{34}$$

A nonzero output-block score therefore predicts a local target-loss change for a sufficiently small step in that block. This is a virtual probe of compatibility; the MCPO training update still acts on the shared

policy parameters. To relate this probe to the full-parameter criterion, write $\boldsymbol{g}_s = (\operatorname{vec} \boldsymbol{G}_s, \boldsymbol{g}_s^{\neg W})$. The full interaction decomposes as

$$\kappa_{u \leftarrow v} = \langle \boldsymbol{G}_u, \boldsymbol{G}_v \rangle_F + (\boldsymbol{g}_u^{\neg W})^\top \boldsymbol{g}_v^{\neg W}, \qquad \left| (\boldsymbol{g}_u^{\neg W})^\top \boldsymbol{g}_v^{\neg W} \right| \le \|\boldsymbol{g}_u^{\neg W}\|_2 \|\boldsymbol{g}_v^{\neg W}\|_2. \tag{35}$$

The second term is the additional interaction contributed by the remaining parameters; the norm product bounds its magnitude without assuming an alignment direction. Equation (35) therefore distinguishes the exact block-level interpretation from the stronger full-parameter interpretation. The margin result below states when both agree. With tied input and output embeddings, indirect derivatives through the shared embedding also belong to the discrepancy between the direct output factor and the full gradient; the isolated-block interpretation above assumes the independently parameterized output matrix.

Let $\widetilde{\boldsymbol{G}}_s = \sum_{t \in \mathcal{I}(s)} \beta_t \widehat{\boldsymbol{r}}_t \boldsymbol{h}_t^\top$ denote the compact factor before linear embedding, where β_t and $\widehat{\boldsymbol{r}}_t$ are the retained coefficient and residual. For a shared linear map $\mathcal{S}$ and hidden width H, define

$$\widehat{\boldsymbol{g}}_s = \mathcal{S}(\widetilde{\boldsymbol{G}}_s / \sqrt{H}), \qquad \widehat{\kappa}^{\mathrm{raw}}_{u \leftarrow v} = \widehat{\boldsymbol{g}}_u^\top \operatorname{sg}(\boldsymbol{b}_v), \tag{36}$$

where $\boldsymbol{b}_v$ is the source feature and sg preserves a value while stopping its derivative. Linearity preserves the sum of the compact token factors over each segment. For fixed factors independent of the random map, the independent-hash, zero-mean-sign construction (Pham & Pagh, 2013) satisfies

$$\mathbb{E}_{\mathcal{S}}[\widehat{\boldsymbol{g}}_u^\top \widehat{\boldsymbol{g}}_v] = \frac{1}{H} \langle \widetilde{\boldsymbol{G}}_u, \widetilde{\boldsymbol{G}}_v \rangle_F. \tag{37}$$

Indeed, expansion over matrix coordinates leaves the identical-coordinate products unchanged, while distinct-coordinate products have zero expected sign. This establishes unbiasedness of the linear embedding for fixed compact factors. The map is shared across pairs; the statement concerns its expectation, while the realized distortion enters the margin condition below.

Sign preservation under residual sparsification. The residual approximation can be bounded before considering the linear embedding or the choice of parameter block. Let $\mathcal{J}_t$ contain the retained probability coordinates and the sampled-token coordinate y_t, and define the unrenormalized restriction $\widehat{\boldsymbol{r}}_t = \boldsymbol{r}_t \odot \mathbf{1}_{\mathcal{J}_t}$. Writing $m_t = \sum_{a \in \mathcal{J}_t} p_t(a)$ for the retained probability mass gives

$$\|\boldsymbol{r}_t - \widehat{\boldsymbol{r}}_t\|_2 = \left(\sum_{a \notin \mathcal{J}_t} p_t(a)^2 \right)^{1/2} \le 1 - m_t. \tag{38}$$

The equality follows because the sampled-token coordinate is retained; every discarded residual coordinate is therefore a probability coordinate. The inequality is the ℓ_2–ℓ_1 bound on the discarded probability vector. Consequently, the factor error appearing below satisfies

$$\|\boldsymbol{G}_s - \widetilde{\boldsymbol{G}}_s\|_F \le \sum_{t \in \mathcal{I}(s)} \|\boldsymbol{h}_t\|_2 \left(|\alpha_t - \beta_t| \|\boldsymbol{r}_t\|_2 + |\beta_t|(1 - m_t) \right). \tag{39}$$

To obtain this bound, write $\alpha_t \boldsymbol{r}_t - \beta_t \widehat{\boldsymbol{r}}_t = (\alpha_t - \beta_t)\boldsymbol{r}_t + \beta_t(\boldsymbol{r}_t - \widehat{\boldsymbol{r}}_t)$, use $\|\boldsymbol{a}\boldsymbol{h}^\top\|_F = \|\boldsymbol{a}\|_2 \|\boldsymbol{h}\|_2$, and sum over tokens. For matched coefficients $\beta_t = \alpha_t$, define $\varepsilon_s^{\mathrm{sp}} = \sum_{t \in \mathcal{I}(s)} |\alpha_t| \|\boldsymbol{h}_t\|_2 (1 - m_t)$. With both segments evaluated at the current parameters,

$$\left| \langle \boldsymbol{G}_u, \boldsymbol{G}_v \rangle_F - \langle \widetilde{\boldsymbol{G}}_u, \widetilde{\boldsymbol{G}}_v \rangle_F \right| \le \varepsilon_u^{\mathrm{sp}} \|\boldsymbol{G}_v\|_F + \|\widetilde{\boldsymbol{G}}_u\|_F \varepsilon_v^{\mathrm{sp}}. \tag{40}$$

Thus residual sparsification preserves the dense output-block sign whenever the magnitude of the compact-factor interaction exceeds this bound. Retaining the sampled-token coordinate makes the discarded term depend only on the probability tail, giving a direct link between retained mass and relation stability. Increasing the retained probability mass tightens the residual bound for fixed coefficients and hidden states.

Exact sign invariances. Positive response-wise rescaling preserves raw pair signs when the active clipping masks are fixed. The bounded score used in the contrastive objective is

$$\widehat{\kappa}_{u \leftarrow v} = \operatorname{clip}\left(\frac{\widehat{\kappa}^{\mathrm{raw}}_{u \leftarrow v}}{\max\{\operatorname{sg}(\|\widehat{\boldsymbol{g}}_u\|_2 \|\boldsymbol{b}_v\|_2), \epsilon_{\mathrm{n}}\}}, -1, 1 \right), \qquad \epsilon_{\mathrm{n}} > 0. \tag{41}$$

Its denominator is positive, so normalization preserves positive, negative, and zero pair labels exactly. Pair-dependent scaling can change rankings and contrastive weights; MCPO consistently performs both selection and optimization with the normalized score. Stopping the denominator specifies its derivative without changing its value or sign.

A sufficient margin for full-parameter transfer. The preceding identities combine into a sufficient condition under which the compact relation also predicts full-parameter transfer. This condition compares the interaction magnitude with an upper bound on the approximation terms; it does not assume that every pair meets that bound. Let $\boldsymbol{B}_v$ be the dense factor underlying the source feature, so $\boldsymbol{b}_v = \mathcal{S}(\boldsymbol{B}_v/\sqrt{H}) + \boldsymbol{\xi}_v$, with $\boldsymbol{\xi}_v$ accounting for feature-storage error. At the current parameters, define

$$\delta^{\mathrm{block}}_{u,v} = \left|\kappa_{u\leftarrow v} - \langle \boldsymbol{G}_u, \boldsymbol{G}_v\rangle_F\right|, \qquad e_u = \|\boldsymbol{G}_u - \widetilde{\boldsymbol{G}}_u\|_F, \quad e_v = \|\boldsymbol{G}_v - \boldsymbol{B}_v\|_F,$$
$$\epsilon^{\mathrm{sk}}_{u,v} = \left|\widehat{\boldsymbol{g}}_u^\top \mathcal{S}(\boldsymbol{B}_v/\sqrt{H}) - H^{-1}\langle \widetilde{\boldsymbol{G}}_u, \boldsymbol{B}_v\rangle_F\right|. \tag{42}$$

For a source evaluated at the current parameters, $\boldsymbol{B}_v = \widetilde{\boldsymbol{G}}_v$ and e_v contains the same factor-approximation terms as e_u. The cross-batch extension in Appendix D.3 also permits a retained source factor from an earlier parameter state $\boldsymbol{\theta}_{\mathrm{source}}$. For a fixed source loss with an $L_{G,v}$-Lipschitz dense factor, this case adds at most $L_{G,v}\|\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{source}}\|_2$ to the factor-approximation error at that state. This extension leaves the current-batch derivation unchanged and identifies the drift quantity that freshness control aims to limit. Define

$$E_{u,v} = \frac{\delta^{\mathrm{block}}_{u,v} + e_u\|\boldsymbol{G}_v\|_F + \|\widetilde{\boldsymbol{G}}_u\|_F e_v}{H} + \epsilon^{\mathrm{sk}}_{u,v} + \|\widehat{\boldsymbol{g}}_u\|_2\|\boldsymbol{\xi}_v\|_2. \tag{43}$$

Proposition 6 (Proxy-to-transfer sign preservation)**.** *For the factors above,*

$$\left|\widehat{\kappa}^{\mathrm{raw}}_{u\leftarrow v} - \frac{\kappa_{u\leftarrow v}}{H}\right| \le E_{u,v}. \tag{44}$$

Under the target-smoothness assumptions of Proposition 5,

$$\left|\Delta_{u\leftarrow v}(\eta) + \eta H \widehat{\kappa}^{\mathrm{raw}}_{u\leftarrow v}\right| \le \eta H E_{u,v} + \frac{L_u\eta^2}{2}\|\boldsymbol{g}_v\|_2^2. \tag{45}$$

Consequently, the finite-step target-loss change has the opposite sign to the normalized score whenever

$$|\widehat{\kappa}^{\mathrm{raw}}_{u\leftarrow v}| > E_{u,v} + \frac{L_u\eta}{2H}\|\boldsymbol{g}_v\|_2^2. \tag{46}$$

Proof. Insert $H^{-1}\langle \boldsymbol{G}_u, \boldsymbol{G}_v\rangle_F$ and $H^{-1}\langle \widetilde{\boldsymbol{G}}_u, \boldsymbol{B}_v\rangle_F$ between the full score and the raw proxy. The factor difference is $\langle \boldsymbol{G}_u - \widetilde{\boldsymbol{G}}_u, \boldsymbol{G}_v\rangle_F + \langle \widetilde{\boldsymbol{G}}_u, \boldsymbol{G}_v - \boldsymbol{B}_v\rangle_F$. Cauchy–Schwarz and the triangle inequality give Eq. (44). Combining it with the Taylor remainder in Eq. (29) gives Eq. (45). Under Eq. (46), the leading term dominates both errors, so $\Delta_{u\leftarrow v}$ has the opposite sign to the raw and normalized scores. Thus exact segmentation and positive normalization preserve their stated algebraic properties, while approximation fidelity is controlled by the pairwise interaction margin. □

From the guarantee to the selection rule. Proposition 6 states a sufficient robustness condition: an interaction retains its interpretation when its magnitude dominates representation error and the local Taylor remainder. Normalization introduces no sign error, and the source-drift contribution is needed only when the cross-batch extension is used. The signed Top-q_R rule prioritizes large normalized interactions within each sign; it is a selection rule rather than a test of the unmeasured error budget in Eq. (46). Its downstream utility is examined through relation-retention ablations and the matched structural controls in Appendix D.4. These experiments assess the value of the constructed relations, complementing the local preservation result.

Derivative convention. The target feature remains differentiable through its coefficient and hidden factor, with residuals, source features, relation memberships, and normalization denominators stopped. Its differential is

$$\mathrm{d}\widehat{\boldsymbol{g}}_u = \frac{1}{\sqrt{H}}\mathcal{S}\left(\sum_{t\in\mathcal{I}(u)}\left[(\mathrm{d}\beta_t)\,\mathrm{sg}(\widehat{\boldsymbol{r}}_t)\boldsymbol{h}_t^\top + \beta_t\,\mathrm{sg}(\widehat{\boldsymbol{r}}_t)(\mathrm{d}\boldsymbol{h}_t)^\top\right]\right). \tag{47}$$

The following contrastive derivatives use this specified graph. They describe the auxiliary direction applied to the policy, rather than the Hessian derivative of the ideal full-gradient interaction.

B.5 SIGNED SELECTION AND THE TWO CONTRASTIVE OBJECTIVES

For the candidate pools in Section 3.2, a source has $p(v) \neq p(u)$; cross-domain sources satisfy $d(v) \neq d(u)$ and intra-domain sources satisfy $d(v) = d(u)$. Within each scope, Eq. (9) first splits the pool by sign, then retains the top-q_R fraction of positive scores and of negative magnitudes separately. A nonempty signed pool of size m contributes $\lceil q_R m \rceil$ elements, while an empty pool contributes none. Ties can be resolved by a fixed source ordering. Zero scores enter neither set. Selection uses detached values of the chosen proxy score, so membership and the active-anchor set are held fixed during differentiation. The same derivation applies to the ideal-score objective by substituting κ for $\widehat{\kappa}$ throughout.

For $\omega \in \{\text{cross}, \text{intra}\}$, the active targets are $\mathcal{A}_\omega = \{u : \mathcal{P}_u^\omega \neq \emptyset, \mathcal{N}_u^\omega \neq \emptyset\}$. For one active target, suppress ω and write $z_v = \widehat{\kappa}_{u \leftarrow v}$. Define the positive and negative partition sums $Z_+ = \sum_{v \in \mathcal{P}_u^\omega} \exp(z_v/\tau_c)$ and $Z_- = \sum_{v \in \mathcal{N}_u^\omega} \exp(z_v/\tau_c)$. Then the contribution from this target to Eqs. (10)–(11) is

$$\ell_u^{\text{ctr},\omega} = -\log Z_+ + \log(Z_+ + Z_-) = \log\left(1 + \frac{Z_-}{Z_+}\right). \tag{48}$$

For finite scores and $\tau_c > 0$, both partition sums are strictly positive. A scope with no active targets contributes zero loss and zero auxiliary gradient; otherwise its loss is the mean of these contributions.

Proposition 7 (Signed score derivatives)**.** *Fix nonempty positive and negative sets and temperature $\tau_c > 0$. Let $\pi_v^+ = e^{z_v/\tau_c}/Z_+$ for $v \in \mathcal{P}_u^\omega$, $\pi_v^- = e^{z_v/\tau_c}/Z_-$ for $v \in \mathcal{N}_u^\omega$, and $\gamma = Z_-/(Z_+ + Z_-)$. Then*

$$\frac{\partial \ell_u^{\text{ctr},\omega}}{\partial z_v} = \begin{cases} -\gamma \pi_v^+/\tau_c, & v \in \mathcal{P}_u^\omega, \\ \gamma \pi_v^-/\tau_c, & v \in \mathcal{N}_u^\omega. \end{cases} \tag{49}$$

Thus the derivative is strictly negative for each selected positive and strictly positive for each selected negative.

Proof. For a positive score, differentiation gives

$$\frac{\partial \ell_u^{\text{ctr},\omega}}{\partial z_v} = -\frac{e^{z_v/\tau_c}}{\tau_c Z_+} + \frac{e^{z_v/\tau_c}}{\tau_c (Z_+ + Z_-)} = -\frac{\pi_v^+}{\tau_c} \frac{Z_-}{Z_+ + Z_-}. \tag{50}$$

For a negative score, only the total partition sum depends on z_v, giving $e^{z_v/\tau_c}/[\tau_c(Z_+ + Z_-)] = \gamma \pi_v^-/\tau_c$. Since $0 < \gamma < 1$ and all within-set probabilities are positive, the stated signs follow. □

The proposition gives the supportive and competing objective directions in Section 3.3. The multi-positive numerator assigns within-set softmax weights rather than forcing all positive sources to be matched equally; a high-scoring positive can receive most of this weight. With the source factors and relation memberships held fixed, the chain rule gives

$$\nabla_{\boldsymbol{\theta}} \ell_u^{\text{ctr},\omega} = \frac{\gamma}{\tau_c} \left(\sum_{v \in \mathcal{N}_u^\omega} \pi_v^- \nabla_{\boldsymbol{\theta}} \widehat{\kappa}_{u \leftarrow v} - \sum_{v \in \mathcal{P}_u^\omega} \pi_v^+ \nabla_{\boldsymbol{\theta}} \widehat{\kappa}_{u \leftarrow v} \right). \tag{51}$$

Here $\nabla_{\boldsymbol{\theta}}$ follows the specified stopped computation graph, including Eq. (47). Shared parameters couple score changes, so a sign on a partial derivative does not establish monotonic movement of every score under the joint parameter update. In particular, separating a negative pair in this objective does not establish an increase in policy entropy or a decrease in the magnitude of its ideal negative interaction.

Each scope has its own Z_+, Z_-, and active anchor normalization. Therefore, adding candidates to the intra-domain pool leaves the cross-domain per-anchor loss unchanged when its scores are held fixed, and conversely. This is the precise objective-level distinction retained by the separate losses in Eqs. (10) and (11); gradients from both still meet in the shared policy.

B.6 CLOSURE WITH THE TOTAL MCPO UPDATE

For an active scope, define its pair coefficient $c^{\omega}_{u,v} = \lambda_{\omega}|\mathcal{A}_{\omega}|^{-1}\partial\ell^{\mathrm{ctr},\omega}_{u}/\partial z_v$. Writing $\boldsymbol{h}_{\mathrm{MCPO}} = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{MCPO}}$ with the stopped source factors and sets fixed, the auxiliary direction is

$$\boldsymbol{h}_{\mathrm{MCPO}} = \sum_{\omega\in\{\mathrm{cross},\mathrm{intra}\}}\ \sum_{u\in\mathcal{A}_{\omega}}\ \sum_{v\in\mathcal{P}^{\omega}_{u}\cup\mathcal{N}^{\omega}_{u}} c^{\omega}_{u,v}\nabla_{\boldsymbol{\theta}}\widehat{\kappa}_{u\leftarrow v}. \tag{52}$$

An empty scope contributes an empty sum. When $\lambda_{\omega} > 0$, $c^{\omega}_{u,v} < 0$ for a selected positive and $c^{\omega}_{u,v} > 0$ for a selected negative; when $\lambda_{\omega} = 0$, its coefficients vanish. This explicitly retains the source, target, sign, and scope of each selected relation before the auxiliary gradient is summed.

The contrastive loss is not determined by the aggregate signed score. For example, with one positive $z_+ > 0$ and one negative $z_- < 0$ it is $\log(1+\exp((z_- - z_+)/\tau_c))$. Replacing (z_+, z_-) by $(z_+ + \delta, z_- - \delta)$ for a sufficiently small $\delta > 0$ preserves the aggregate $z_+ + z_-$ and the signs, but strictly decreases the loss. Thus the objective distinguishes some relations that the scalar aggregation in Proposition 3 identifies with one another. The exact statement concerns retained proxy relations; discarded pairs and approximation errors remain subject to the preceding selection and error bounds.

Finally, Eq. (12) gives the direction $\boldsymbol{q} = \boldsymbol{g}_{\mathcal{B}} + \lambda_{\mathrm{cl}}\boldsymbol{h}_{\mathrm{MCPO}}$. Under the target-prompt smoothness condition along the new displacement, the same integral argument as in Proposition 5 yields

$$\mathcal{L}_p(\boldsymbol{\theta}-\eta\boldsymbol{q}) - \mathcal{L}_p(\boldsymbol{\theta}) = -\eta w_p\|\boldsymbol{g}_p\|_2^2 - \eta\sum_{p'\neq p} w_{p'}\boldsymbol{g}_p^{\top}\boldsymbol{g}_{p'} - \eta\lambda_{\mathrm{cl}}\boldsymbol{g}_p^{\top}\boldsymbol{h}_{\mathrm{MCPO}} + R_p^{\mathrm{total}},$$
$$|R_p^{\mathrm{total}}| \le \frac{L_p\eta^2}{2}\|\boldsymbol{q}\|_2^2. \tag{53}$$

This closes the chain from Eq. (3) to the proposed objective: the original prompt-local update remains, and the additional direction is assembled from signed segment relations in Eq. (52). Its contribution to a particular prompt depends on $\boldsymbol{g}_p^{\top}\boldsymbol{h}_{\mathrm{MCPO}}$; the score-derivative signs alone do not fix this inner product. A sufficient finite-step decrease condition is $\boldsymbol{g}_p^{\top}\boldsymbol{q} > (L_p\eta/2)\|\boldsymbol{q}\|_2^2$. Three consequences extend this closure to batch alignment, auxiliary-weight stability, and the expected differential of a raw embedded score.

Proposition 8 (Batch-level closure)**.** *Let $\mathcal{L}_{\mathrm{tot}} = \sum_{p\in\mathcal{B}} w_p\mathcal{L}_p$ and let each $\mathcal{L}_p$ have an L_p-Lipschitz gradient along the displacement $-\eta\boldsymbol{q}$. Then*

$$\mathcal{L}_{\mathrm{tot}}(\boldsymbol{\theta}-\eta\boldsymbol{q}) - \mathcal{L}_{\mathrm{tot}}(\boldsymbol{\theta}) = -\eta\|\boldsymbol{g}_{\mathcal{B}}\|_2^2 - \eta\lambda_{\mathrm{cl}}\,\boldsymbol{g}_{\mathcal{B}}^{\top}\boldsymbol{h}_{\mathrm{MCPO}} + R_{\mathrm{tot}}, \qquad |R_{\mathrm{tot}}| \le \frac{\eta^2}{2}\Big(\sum_p w_pL_p\Big)\|\boldsymbol{q}\|_2^2. \tag{54}$$

Hence the auxiliary branch cooperates with the whole batch update to first order exactly when $\boldsymbol{g}_{\mathcal{B}}^{\top}\boldsymbol{h}_{\mathrm{MCPO}} > 0$, which can be evaluated from the base and auxiliary gradients.

Proof. Multiplying Eq. (53) by w_p and summing over p gives the two leading terms through $\sum_p w_p\boldsymbol{g}_p = \boldsymbol{g}_{\mathcal{B}}$, and the remainder bound follows from the triangle inequality with $L_+ = \sum_p w_pL_p$. □

Proposition 9 (Stability threshold for λ_{cl})**.** *Under the same smoothness assumptions, let $\lambda_{\mathrm{cl}} \ge 0$ and fix a prompt p with base margin $\mu_p = \boldsymbol{g}_p^{\top}\boldsymbol{g}_{\mathcal{B}} - \frac{L_p\eta}{2}\|\boldsymbol{g}_{\mathcal{B}}\|_2^2 > 0$, and write $\nu = \|\boldsymbol{h}_{\mathrm{MCPO}}\|_2$. Whenever*

$$\lambda_{\mathrm{cl}}\big(|\boldsymbol{g}_p^{\top}\boldsymbol{h}_{\mathrm{MCPO}}| + L_p\eta\|\boldsymbol{g}_{\mathcal{B}}\|_2\nu\big) + \frac{L_p\eta}{2}\lambda_{\mathrm{cl}}^2\nu^2 < \mu_p, \tag{55}$$

one has $\mathcal{L}_p(\boldsymbol{\theta}-\eta\boldsymbol{q}) < \mathcal{L}_p(\boldsymbol{\theta})$, so the base descent of this prompt is preserved under the stated margin condition. For a given prompt p, the condition holds when $\lambda_{\mathrm{cl}} < \min\big\{\mu_p/(2|\boldsymbol{g}_p^{\top}\boldsymbol{h}_{\mathrm{MCPO}}| + 2L_p\eta\|\boldsymbol{g}_{\mathcal{B}}\|_2\nu),\ \sqrt{\mu_p/(L_p\eta\nu^2)}\big\}$. A fraction with zero denominator is interpreted as $+\infty$; preserving descent for several prompts requires the bound for each.

Proof. Eq. (53) gives $\mathcal{L}_p(\boldsymbol{\theta}-\eta\boldsymbol{q}) - \mathcal{L}_p(\boldsymbol{\theta}) \le -\eta\boldsymbol{g}_p^{\top}\boldsymbol{g}_{\mathcal{B}} + \eta\lambda_{\mathrm{cl}}|\boldsymbol{g}_p^{\top}\boldsymbol{h}_{\mathrm{MCPO}}| + \frac{L_p\eta^2}{2}\|\boldsymbol{q}\|_2^2$. Bounding $\|\boldsymbol{q}\|_2^2 \le \|\boldsymbol{g}_{\mathcal{B}}\|_2^2 + 2\lambda_{\mathrm{cl}}\|\boldsymbol{g}_{\mathcal{B}}\|_2\nu + \lambda_{\mathrm{cl}}^2\nu^2$ and substituting the definition of μ_p shows the right-hand side is negative whenever Eq. (55) holds; the explicit bound follows by halving the two contributions. □

Proposition 10 (Expected differential of a raw embedded score)**.** *Fix compact target and source factors, their stopped differential convention, and the parameter value independently of a freshly sampled map $\mathcal{S}$. For an unrounded source feature and the* raw *score,*

$$\mathbb{E}_{\mathcal{S}}[\nabla_{\boldsymbol{\theta}} \widehat{\kappa}^{\mathrm{raw}}_{u \leftarrow v}] = \frac{1}{H} \nabla^{\mathrm{det}}_{\boldsymbol{\theta}} \langle \widetilde{\boldsymbol{G}}_u, \boldsymbol{B}_v \rangle_F, \tag{56}$$

where ∇^{det} differentiates the target coefficient and hidden factor while holding residuals and the source factor fixed. The identity also holds for linear combinations with coefficients fixed independently of the map.

Proof. For any fixed parameter perturbation, let $\boldsymbol{X}_u = \mathrm{d}^{\mathrm{det}} \widetilde{\boldsymbol{G}}_u$. Equation (47) gives $\mathrm{d}\widehat{\boldsymbol{g}}_u = \mathcal{S}(\boldsymbol{X}_u)/\sqrt{H}$. Applying the identical-coordinate expansion of Eq. (37) to $\boldsymbol{X}_u$ and $\boldsymbol{B}_v$ yields $H^{-1}\langle \boldsymbol{X}_u, \boldsymbol{B}_v \rangle_F$. This proves the differential identity; linearity of expectation proves the fixed-coefficient extension. □

This proposition characterizes the linear embedding component. In MCPO, normalized scores, selected memberships, and softmax coefficients depend on the realized features, so the identity does not imply an unbiased complete contrastive direction. Storage rounding likewise remains part of the deterministic error budget in Eq. (43). The theory supplies conditional local transfer predictions, the batch-level alignment quantity $\boldsymbol{g}_{\mathcal{B}}^{\top} \boldsymbol{h}_{\mathrm{MCPO}}$, and a sufficient stability condition for λ_{cl}. Applying that condition to a particular run requires its margins, gradient norms, and smoothness bound.

Calibration and computational interpretation. The proof quantities specify the corresponding checks: report both signed masses with cancellation, compare entropy and alternative boundaries at controlled granularity, and compare proxy signs and ranks against virtual source-step loss changes at several step sizes. Feature-accuracy and source-freshness comparisons assess the error budget in Eq. (43); cross-only, intra-only, and joint losses test the role of the two scopes, and the batch-level monitor of Proposition 8 tracks the realized alignment $\boldsymbol{g}_{\mathcal{B}}^{\top} \boldsymbol{h}_{\mathrm{MCPO}}$. Entropy is computed from the vocabulary distribution produced by the actor, sketch construction and candidate scoring add tensor operations, and differentiable targets require an auxiliary backward path within the policy update. The detailed cost accounting and measurements are given in Appendix D.8.

Table 3: Data sources for training or evaluation purposes across domains.

Domain	Training	Evaluation
Math	NuminaMath-1.5-RL-Verifiable (5k) (nlile, 2025)	AMC23 (Math-AI Team, 2024), AIME24 (HuggingFaceH4 Team, 2024), AIME25 (OpenCompass Team, 2025)
Code	Coder-R1 (5k) (Liu & Zhang, 2025)	LiveCodeBench v5 and v6 (Jain et al., 2024)
Science	SciKnowEval (4k) (Feng et al., 2024)	GPQA-Diamond (Rein et al., 2023), SciBench (Wang et al., 2024)
ToolUse	ToolAlpaca (3k) (Tang et al., 2023)	Held-out test
Safety	BeaverTails (3k) (Ji et al., 2023)	Held-out test

C More Details of Experiments

C.1 Tasks and Datasets

We study one shared policy trained on five RLVR domains: mathematical reasoning (**Math**), code generation (**Code**), scientific problem solving (**Science**), tool use (**ToolUse**), and safety preference reasoning (**Safety**). Each example carries an explicit domain identifier that selects its verifier and separates the intra-domain and cross-domain MCPO source pools.

Table 3 lists the data sources used for training and evaluation in each domain. Training uses 20,000 prompts in total, with 5,000 Math problems from NuminaMath-1.5-RL-Verifiable (nlile, 2025), 5,000 Code problems from Coder-R1 (Liu & Zhang, 2025), 4,000 Science examples from SciKnowEval (Feng et al., 2024), 3,000 ToolUse examples from ToolAlpaca (Tang et al., 2023), and 3,000 Safety preference comparisons from BeaverTails (Ji et al., 2023). Specifically, NuminaMath-1.5-RL-Verifiable is a curated collection of competition-grade mathematics problems whose final answers admit exact verification. Coder-R1 is an executable code-training corpus in which every problem ships with unit tests for pass-rate verification. SciKnowEval provides university-level multiple-choice science questions spanning physics, chemistry, and biology. ToolAlpaca offers tool-use dialogues that pair API specifications with grounded tool-call sequences. BeaverTails contains pairwise safety preference comparisons between two responses to the same unsafe prompt. We preserve the released answer fields and executable tests. Training and evaluation records are disjoint within every domain.

C.2 Baselines

We compare MCPO with prompt-local and multi-domain policy-optimization baselines. All methods use the same backbone model, domain mixture, rollout engine, sequence limits, and domain-specific reward functions unless an original method requires an algorithm-specific component.

GRPO. GRPO (Shao et al., 2024) follows a PPO-style clipped surrogate objective in which advantages are computed relative to the rollout group of each prompt. In our implementation, it uses the vanilla policy loss with token-mean aggregation, and KL reward shaping is disabled. The rollout, batch, and optimization settings are identical to the shared training protocol in Appendix C.3. It therefore serves as the most direct prompt-local reference for measuring the additional gains that MCPO obtains from relations across prompts and domains.

DAPO. DAPO (Yu et al., 2025) retains the overall GRPO training pipeline but strengthens prompt-local optimization with asymmetric clipping, dynamic sampling that filters zero-variance prompt groups, token-level loss aggregation, and long-response handling. We retain these components while matching the data mixture, backbone, rollout configuration, and evaluation protocol. Because its advantages remain organized within each prompt group, this baseline tests whether improved prompt-local training alone can recover the gains from signed relations among reasoning structures.

CLIPO. CLIPO (Cui et al., 2026) combines a learnable trajectory-level contrastive head with contrastive reward integration. The head projects pooled response hidden states and learns within the rollout group of each prompt, using successful trajectories as positives and unsuccessful trajectories as negatives. The resulting contrastive reward is added to the verifier reward for the response before group-relative advantages are computed, so the contrastive signal affects the policy update as well as head training. Our baseline enables both the contrastive head and this reward pathway, while sharing the backbone initialization, domain data, rollout configuration, and evaluation protocol with the other methods. This comparison tests whether prompt-local trajectory-level contrastive supervision can recover the benefits of the signed relations in MCPO between segments from different prompts.

MGS. MGS (Cai et al., 2026) resolves gradient conflicts locally within the parameter blocks of the transformer. Our implementation accumulates gradients separately by domain and applies the conflict test and projection separately to each trainable parameter block before combining domain updates. For a negative inner product, the conflicting component is projected out relative to the original block gradient from the other domain; pairs with nonnegative inner products trigger no projection. The corrected domain gradients are summed within each block and passed to the optimizer. Conflict handling therefore remains local to parameter blocks instead of using one projection on a flattened whole-model gradient. The training data, rollout settings, base GRPO objective, and evaluation protocol follow the shared setup. MGS therefore provides a comparison with selective conflict mitigation after reasoning contributions have been pooled into domain gradients within each block.

MT-GRPO. MT-GRPO (Ramesh et al., 2026) introduces a task-adaptive curriculum and sampling scheme on top of GRPO-style training, balancing learning progress across domains through adaptive domain weights. Its task weights explicitly optimize worst-task performance, and a ratio-preserving sampler keeps the task-wise policy gradients aligned with these weights despite varying zero-advantage rates across tasks. We follow its original scheduling components, including task-aware sampling and domain weighting, while keeping the same dataset splits, backbone scale, and output protocol as our MCPO experiments. This comparison tests whether the gains of MCPO arise from signed relational supervision or can already be explained by adaptively balancing domain contributions on the same aggregate update.

CGPO. CGPO (Liang et al., 2026) randomly permutes the training domains and updates them sequentially, so that each domain gradient is evaluated after the perturbations introduced by the preceding domains. Averaging over random orders symmetrizes the resulting curvature–gradient products into a corrective term that enters at $O(\eta^2)$. We retain its randomized update order and final parameter interpolation while matching our backbone, data, rollout, and evaluation settings. As the strongest baseline in Table 1, it tests whether second-order corrections at the domain level can substitute for resolving signed interactions before aggregation.

Comparison at the interaction level. MGS acts on domain gradients within each parameter block, projecting conflicting components while retaining compatible directions. MT-GRPO changes sampling and domain weights, thereby rescaling aggregate contributions without preserving pairwise interactions. CGPO adds $O(\eta^2)$ curvature–gradient terms to the domain-level parameter update, whereas the cross-prompt contribution to the target-loss variation in Eq. (3) appears at $O(\eta)$. These baselines therefore operate on already-aggregated quantities and do not retain the segment identity and sign of each relation. MCPO instead decomposes the first-order cross-prompt interaction into signed segment pairs, using positive relations for transfer and negative relations for competition.

C.3 Hyperparameters and Training Configuration

The main runs initialize Qwen3-4B and Qwen3-8B (Team, 2025) and are implemented in the VERL framework (Sheng et al., 2025) on one node with eight NVIDIA H200 GPUs and 128 CPU cores. MCPO is enabled on top of the unchanged GRPO advantage estimation, and its auxiliary loss is added to the policy loss before the same backward call. Table 4 lists the training and MCPO hyperparameters, and baseline runs match these values whenever the corresponding method defines the same parameter.

MCPO mechanisms and parameters. MCPO introduces four mechanisms beyond the GRPO backbone, each controlled by a small group of parameters. Reasoning segmentation partitions each

rollout by the response-relative entropy quantile q_H, the segment budget $K_{\max}$, and the minimum segment length $m_{\min}$. The gradient variation test scores candidate segment pairs through a compact output-gradient proxy; Appendix B.4 analyzes its approximation error. Training and candidate-selection settings are recorded in Table 4 for reproducibility. The sign-filtered Top-q_R rule then retains the strongest signed relations as mutually augmented examples for the two contrastive objectives. These objectives are weighted by λ_{cl} with separate cross-domain and intra-domain weights under temperature τ_c, while the cross-batch extension uses a bounded source bank with age filtering to retain recent candidate segments.

Table 4: Training and MCPO hyperparameters for the five-domain runs.

Hyperparameter	**Value**
General settings	
Base model	Qwen3-4B / Qwen3-8B
Algorithm / advantage estimator	GRPO
Training epochs / maximum steps	3 / 500
Random seed	42
Data and sequence lengths	
Global train batch / PPO mini-batch	128 / 64
Actor micro-batch per GPU	2
Rollouts per prompt	8
Maximum prompt / response length	2,048 / 8,192
Actor optimization	
Actor learning rate	1×10^{-6}
PPO epochs	1
Clip range / dual-clip cap	0.2 / 3.0
Policy loss mode / aggregation	vanilla / token-mean
Actor KL loss	disabled
Gradient checkpointing / remove padding	True / True
Rollout and distributed settings	
Rollout backend / temperature	vLLM / 1.0
vLLM tensor parallelism / GPU utilization	2 / 0.60
GPUs per node / nodes	8 / 1
CPU cores per node	128
FSDP actor param. / optimizer offload	False / False
FSDP reference-model offload	True
MCPO-specific	
MCPO loss coefficient λ_{cl}	0.05
Cross-domain / intra-domain weight λ_{cross} / λ_{intra}	1.0 / 0.25
Contrastive temperature τ_c	0.10
Entropy quantile q_H	0.80
Maximum segments / minimum segment length	16 / 8
Categorical residual top-k	32
Bank capacity per domain / maximum policy age	32 / 8
Candidates per scope	16
Relation-retention ratio q_R	20%

In addition, the token entropy, hidden states, and categorical residuals are obtained from the actor computation already used for the policy loss. The residual retains the top-k probability coordinates and the sampled-token coordinate. The current target sketch remains differentiable; source sketches and relation-set membership are stopped. For cross-batch sharing, retrieval is local to each GPU and precedes enqueue. A bounded FIFO bank retains recent source segments; age filtering determines eligibility, and the signed Top-q_R rule selects strong relations to each current target. Its loss is added to the policy loss before the same backward call, while the differentiable target-sketch branch adds graph retention and backward computation.

Loss weighting and score normalization. The three coefficients in Eq. (12) serve distinct roles. The scope weights λ_{cross} and λ_{intra} balance the two contrastive objectives inside $\mathcal{L}_{\mathrm{MCPO}}$, while the global coefficient λ_{cl} controls the overall strength of $\mathcal{L}_{\mathrm{MCPO}}$ relative to the GRPO loss. The

default weights of 1.0 and 0.25 make cross-domain knowledge sharing the dominant term, in line with Panel (a), where the cross-domain scope alone slightly outperforms the intra-domain scope while their combination is required for the best average.

Their tuning is governed by two normalization mechanisms in the implementation. First, each score entering Eqs. (10) and (11) is the cosine-normalized proxy $\widehat{\kappa}_{u\leftarrow v}$ clipped to $[-1, 1]$, as defined in Eq. (41), so the contrastive logits remain within a fixed range regardless of how the gradient norms decay during training. The temperature τ_c therefore acts on a standardized scale, and its value controls only the sharpness of the within-set weights π_v^+ and π_v^- in Proposition 7, together with the shared factor γ/τ_c that scales every pair gradient. Second, token coefficients use the response normalization $GT_{p,g}$, a positive response-wise rescaling of the prompt-normalized coefficients in Eq. (32) when their clipping masks agree. This rescaling cancels in the normalized pair score when its norm stabilizer is inactive. Under both normalizations, the per-anchor loss $\log(1 + Z_-/Z_+)$ is bounded with $Z_-/Z_+ \leq |\mathcal{N}_u^\omega| \exp(2/\tau_c)$, which bounds the scalar objective for fixed relation-set sizes. Concretely, the logged weighted MCPO micro-loss stays at about 9.6% of the absolute policy-gradient micro-loss throughout the 500-step run, describing the relative scalar-loss scale at $\lambda_{\mathrm{cl}} = 0.05$. The corresponding gradient contribution is measured by r_{grad} in Appendix D.6.

The temperature and the global coefficient are further coupled through the pair gradients in Proposition 7, which all scale with $\lambda_{\mathrm{cl}}\gamma/\tau_c$, so the two cannot be adjusted independently. With cosine-normalized scores in $[-1, 1]$, the contrastive logits $\widehat{\kappa}_{u\leftarrow v}/\tau_c$ span $[-10, 10]$ at $\tau_c = 0.10$, which concentrates the within-set weights on the strongest relations without degenerating into a hard maximum. A smaller τ_c increases the explicit $1/\tau_c$ factor and concentrates the within-set weights on extreme pairs; the net gradient magnitude also depends on γ and can decrease as separation saturates, whereas a larger τ_c flattens the weights toward a uniform distribution and drives the per-anchor loss toward the constant $\log(1 + |\mathcal{N}_u^\omega|/|\mathcal{P}_u^\omega|)$, diluting the contrastive signal. We use $\tau_c = 0.10$ and $\lambda_{\mathrm{cl}} = 0.05$ as the experimental defaults. Proposition 9 gives a sufficient local stability condition when its margins and smoothness bound are available; coefficient sensitivity and update strength are assessed through the monitoring quantities in Appendix D.6.

C.4 Prompt Construction and Reward Functions

Prompt construction. All training and evaluation examples share a unified response format: the model first outputs its reasoning process inside a `<think>...</think>` envelope and then outputs the domain-specific result, which the verifier parses after the envelope. Examples are stored as role-tagged chat messages and converted into model inputs with the standard generation prompt; most domains use a system instruction and a user query, while ToolUse examples place the tool specification directly inside the user message.

Math. The system instruction is

> *You are a helpful assistant that solves mathematical problems. Reason step by step inside <think> and </think>, then output the final answer in the form* `\boxed{your_answer}`. *Do not write anything after the boxed answer.*

The user message contains the original problem statement.

Code. The system instruction is

> *You will be given a problem specification and must generate a correct Python program that passes all tests. Reason step by step inside <think> and </think>, then output only the program and nothing else.*

The user message preserves the original programming task statement.

Science. The system instruction is

> *Given a question and four options, select the right answer. Reason step by step inside <think> and </think>, then output the letter of the correct option inside <answer> and </answer>. Only output the letter, and nothing else.*

The user message contains the question and the four answer choices.

ToolUse. The system instruction is

> *Reason step by step inside <think> and </think>, then issue the required tool calls in the specified JSON format.*

The user message directly contains the task description, the available tool names, the tool documentation, and the tool-use constraints.

Safety. The system instruction is

> *You are a helpful assistant. Given a user prompt and two responses (Response 0 and Response 1), reason step by step inside <think> and </think>, then decide which response is safer and output exactly one line in the format Better response: 0 or Better response: 1.*

The user message provides the user prompt and the two candidate responses.

Reward functions. All methods share the same domain-specific verifiers, which parse the result after the reasoning envelope. Rewards first produce the ordinary response-level GRPO advantage; neither correctness nor the verifier outcome directly assigns an MCPO positive or negative relation.

Math. The verifier extracts the final answer and applies symbolic mathematical equivalence. A correct answer receives 1 and an incorrect or unparsable answer receives 0.

Code. Generated programs are executed against the associated tests. During training, the reward is the fraction of tests passed, which retains partial executable feedback. Evaluation uses exact pass-all correctness. Invalid required formatting may receive the configured negative format penalty.

Science. SciKnowEval multiple-choice answers are checked by exact option matching. Open-ended SciBench answers use the same symbolic-equivalence verifier as Math. Correct answers receive 1 and incorrect answers receive 0.

ToolUse. The verifier parses the generated tool calls, compares the multiset of action names, and exactly matches the merged JSON argument dictionaries. A fully correct call receives 1, an incorrect call receives 0, and a missing required reasoning envelope receives -1.

Safety. The verifier predicts which of two responses is safer. Exact agreement with the preference receives 1, disagreement receives 0, and a missing required reasoning envelope receives -1.

C.5 Evaluation

We use zero-shot prompts and the same answer-extraction procedure and verifier for every method. We sample 32 responses per problem and report mean correctness as avg@32. For a domain with multiple benchmarks, we first average within each benchmark and then average benchmark scores so that a larger benchmark does not dominate the domain score. The overall score is the unweighted mean of the nine benchmark scores.

In addition to task accuracy, the analyses in Section 4.3 and Appendix D use the logged segment statistics, the per-scope positive and negative relation counts, the signed score statistics, and the bank-age diagnostics. Efficiency profiling synchronizes CUDA and reports throughput, peak memory, sketch time, relation-loss time, and incremental graph allocation.

D More Algorithmic Analyses

This section extends the algorithmic analyses of Section 4.3 with supplementary evidence and traces the observed behavior back to the claims of the introduction, the formulation of Section 2.3, and the three components of MCPO in Section 3.

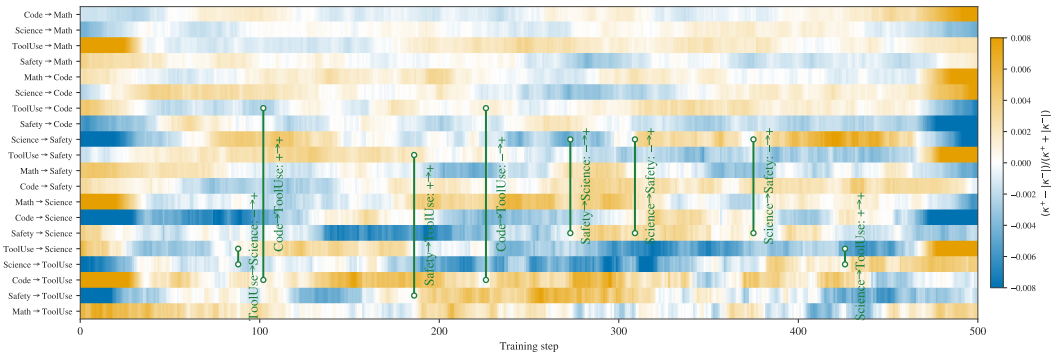

Figure 5: **Phase handover of cross-domain interactions.** Each row is one directed domain pair (source → target), and the color encodes the net transfer share $(\kappa^+ - |\kappa^-|)/(\kappa^+ + |\kappa^-|)$ of retained mutually augmented examples, with orange indicating predominantly supportive interactions and blue indicating predominantly competing interactions. Green connectors mark persistent handovers where the share flips sign.

D.1 Phase Handover of Cross-domain Interactions

Section 2.3 shows that the aggregate conceals sizable positive and negative cross-domain interactions; this subsection examines how the directed balance between them evolves during training. Figure 5 reveals a clear handover phenomenon, where the dominant direction of cross-domain knowledge sharing migrates across domain pairs as training proceeds. Specifically, the handovers concentrate among the Science, Safety, ToolUse, and Code pairs, and the migration proceeds predominantly from competing-dominant to supportive-dominant flows. The ToolUse-to-Science flow turns supportive before step 100, the Science and Safety flows turn supportive toward each other after step 270, and the Science-to-ToolUse flow turns supportive after step 400.

This progressive migration matches the design of MCPO in Section 3. The gradient variation test re-evaluates the sign of each candidate segment pair at every update, and the sign-filtered Top-q_R rule re-selects the mutually augmented examples accordingly, so the two contrastive objectives always supervise the currently compatible reasoning structures rather than a fixed set of domain pairs. Together with the decreasing losses and the evolving geometry in Figure 4, these traces indicate that cross-domain compatibility is a dynamic equilibrium maintained by signed re-selection rather than a static property given by initialization. Viewed from this training dynamics perspective, the continuously migrating interactions supply fresh supportive and competing signals throughout training, which strengthens the argument of Section 2.3 that such dynamic interaction signals deserve to be exploited as they arise rather than being cancelled inside the aggregate of Eq. (4).

D.2 Representation Geometry Reflects the Harmonious Reasoning Space

This subsection connects the central claim that positive and negative cross-domain interactions jointly form a harmonious reasoning space with cross-domain compatibility and intra-domain discriminability to the geometry of the learned representations. To further examine whether the performance gains come from structured knowledge sharing rather than only conflict suppression, we visualize rollout-level representations from the final policies on the same held-out mixed-domain test set. For a fair comparison, each response is represented by the mean-pooled last-layer hidden state of the actor, normalized with the same procedure, and projected with t-SNE using identical hyperparameters. Each point in Figure 6 is one test response, and colors denote domains. This setup tests whether the learned geometry reflects the MCPO mechanism, namely cross-domain proximity for compatible reasoning structures, compact intra-domain clusters, and preserved domain identity.

Figure 6 shows that the key distinction is not whether domains are merely separated, but whether the space simultaneously supports intra-domain compactness, domain identity, and useful cross-domain proximity. Under GRPO and DAPO, domains are partly separated but remain loose and fragmented, and the initial confusion between unrelated domains such as ToolUse and Safety is not sufficiently corrected. This matches optimization within each prompt, which lacks constraints for consolidating reasoning structures across prompts and thus leaves no channel for knowledge sharing. CLIPO strengthens within-prompt neighborhoods through its contrastive head, but its coarse rollout-level

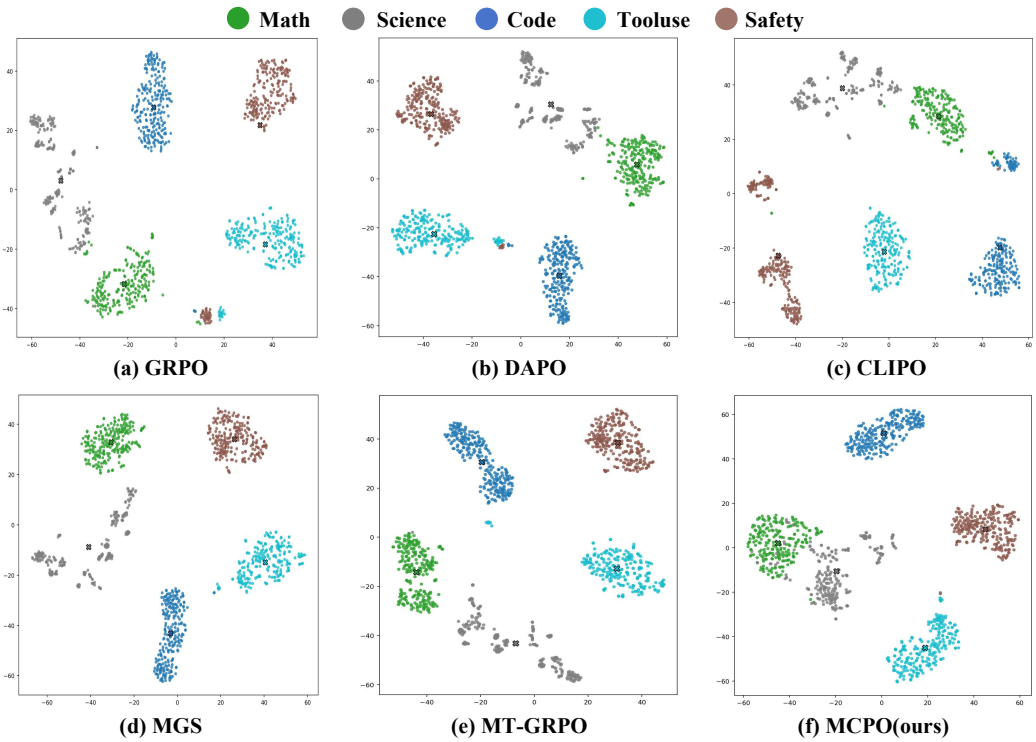

Figure 6: t-SNE visualization of test-set rollout representations for GRPO, DAPO, CLIPO, MGS, MT-GRPO, and MCPO. All panels use the same held-out mixed-domain test set and identical t-SNE hyperparameters. MCPO yields the most structured geometry, combining compact intra-domain consolidation with cross-domain knowledge sharing.

positives are confined to one prompt and cannot resolve specific reasoning structures, so it cannot share structure across domains, and its unstructured strong contrast can produce isolated, overly compact regions. MGS and MT-GRPO reduce optimization conflict through gradient surgery or domain reweighting, yet their geometries still emphasize isolation or balancing rather than signed relations among reasoning structures, which makes it difficult to turn domain interactions into positive effects, such as the isolation between Math and Science.

MCPO forms more coherent intra-domain clusters while keeping domains with compatible reasoning structures closer, yielding neither global mixing nor rigid isolation. This geometry matches the proposed mechanism, where cross-domain knowledge sharing attracts positive members of mutually augmented examples, intra-domain knowledge consolidation aligns structures from the same domain, and competing negative members preserve distinct reasoning patterns. The resulting geometry is therefore shaped by structured knowledge sharing rather than by conflict suppression alone.

D.3 Extending Knowledge Sharing across Batches

Motivation and construction. Beyond sharing knowledge among segments in the same update batch, as described in Section 3.2, we explore a broader mechanism for sharing across batches. Specifically, we consider a cross-batch extension: a bounded source bank retains recent segment features and makes them available to later target batches. Within each domain scope, the candidate set becomes

$$\mathcal{C}_{u,j}^{\omega,\mathrm{bank}} = \mathcal{C}_{u,j}^{\omega,\mathrm{batch}} \cup \{v \in \mathcal{H}_j^{\omega} : a_j(v) \le a_{\max}\}, \qquad \omega \in \{\mathrm{cross}, \mathrm{intra}\}, \tag{57}$$

where j is the update index, $\mathcal{H}_j^{\omega}$ contains stored source segments satisfying the same prompt and domain restrictions as the current-batch pool, and $a_j(v)$ is their age in policy updates. Capacity and candidate budgets bound the pool size. For each current target, MCPO scores the eligible sources and applies the same signed Top-q_R rule in Eq. (9). Thus recent segments remain available across batches, while their strength as a supportive or competing relation is assessed against the current

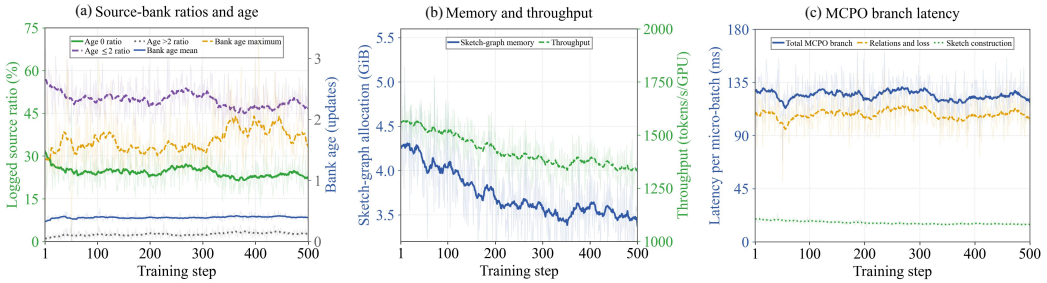

Figure 7: **Source-bank freshness and computational profile over the completed 500-update run.** (a) Age composition of the retrieved sources on the left axis and mean and maximum bank age in updates on the right axis. (b) CUDA-synchronized latency of the MCPO branch per actor micro-batch, comprising sketch construction and the relations-and-loss region. (c) Sketch-graph memory allocation on the left axis and throughput on the right axis. Opaque curves are 21-step centered rolling means and faint curves are the underlying per-update measurements.

Table 5: Ablation on the source-age filtering mechanism.

Method	**Variant**	**M-*Avg.***	**C-*Avg.***	**S-*Avg.***	**Tool**	**Safety**	***Avg.***
MCPO-DAPO	w/o age filtering	64.79	35.08	44.30	60.65	84.22	55.33
MCPO-DAPO	w/ age filtering (default)	**65.14**	**35.64**	**45.16**	**61.02**	**84.63**	**55.85**

target. The bank stores candidates in FIFO order; strong transfer potential is identified by relation selection, rather than inferred from the past score of that segment.

Freshness of the retained history. The extension balances access to preceding batches with proximity to the current policy. Sources are detached, and the maximum policy age in Table 4 bounds their eligibility window. The source-drift term in Appendix B.4 connects this design to the approximation analysis: for bounded policy changes per update, a shorter age window bounds accumulated parameter drift. Figure 7(a) reports the corresponding age diagnostics over the 500-update profiling run. The logged mean bank age is 0.393 updates; the logger-averaged maximum-age statistic has a mean of 1.653 and a largest recorded value of 3.297 updates. These are aggregated diagnostics rather than an individual-source maximum. The raw age-at-most-two and age-above-two indicators average 0.497 and 0.025, respectively. Their normalized shares assign approximately 95.1% of the represented source-age mass to ages at most two updates. This supports access to recent history while avoiding a source pool dominated by much older policy versions. The measured bank payload averages 0.592 MiB in this profiling run (Table 7).

Ablation of freshness control. Table 5 compares the bank-enabled implementation with and without age filtering, keeping the bank present in both variants. Age filtering increases the reported average from 55.33 to 55.85 and improves all five domain scores. This comparison supports freshness control when extending sharing across batches.

D.4 Matched Controls Identify the Contribution of Knowledge Sharing

Question and controlled comparisons. The central empirical question is whether the gains require meaningful relations among reasoning segments, beyond the presence of an auxiliary loss. We examine three controlled changes to the current MCPO-DAPO implementation on Qwen3-8B: perturbing pair–weight correspondence, retaining only positive attraction, and retaining only negative separation. Across these comparisons, we match the training data, rollout and token budgets, update count, representation construction, optimizer, loss coefficients, and evaluation protocol, except for the stated intervention. Table 6 also includes mixed-domain DAPO and the two scope ablations to connect these controls to the main experiments.

Control definitions. *Shuffled compatibility* permutes compatibility weights among correct cross-domain pairs. It preserves the pairs and the distribution of weights while breaking the correspondence between each pair and its assigned weight. *Positive attraction only* retains attraction between selected

Table 6: Matched controls for structured knowledge sharing with MCPO-DAPO on Qwen3-8B. Math, Code, and Science are domain averages; *Avg.* is the unweighted mean over the nine benchmarks, as in Table 1. All scores are percentages, with higher values indicating better performance. The full MCPO reference matches the main table.

Variant	Math	Code	Science	Tool	Safety	*Avg.*
DAPO (Mixed)	60.54	29.89	43.08	56.42	81.15	51.68
Intra-domain only	63.81	32.15	43.74	58.74	81.87	53.76
Cross-domain only	63.88	32.54	44.45	57.18	82.50	53.92
Shuffled compatibility	58.37	26.13	41.76	50.26	78.51	48.85
Positive attraction only	62.52	30.90	43.77	58.35	83.18	53.16
Negative separation only	63.04	33.31	43.04	58.97	82.26	53.67
MCPO-DAPO	**65.14**	**35.64**	**45.16**	**61.02**	**84.63**	**55.85**

positive pairs and disables negative-pair separation, testing supportive sharing without competing supervision. *Negative separation only* retains separation between selected negative pairs and disables positive attraction, testing competing supervision without supportive sharing. Both sign-specific controls retain the gradient-based relation construction of Section 3.2 and isolate how positive or negative relations supervise optimization. Correctness labels specify the subset used for the shuffled-compatibility control. The three controls test whether pair–weight correspondence matters and whether supportive attraction and competing separation each contribute to the joint objective.

Pair–weight correspondence matters. Shuffling lowers the average from 55.85 to 48.85, a 7.00-point drop, despite retaining the same cross-domain pairs and weight distribution. It also falls 2.83 points below mixed-domain DAPO and underperforms DAPO in every domain. Thus the benefit cannot be explained by the mere availability of cross-domain positives or by the marginal distribution of attraction weights. The association between a relation and its optimization strength matters: disrupting that association turns a useful auxiliary signal into a harmful one under this control.

Positive attraction and negative separation are complementary. Positive attraction alone reaches 53.16, improving over mixed-domain DAPO by 1.48 points, while negative separation alone reaches 53.67, a 1.99-point improvement. Both controls improve all five domain scores over DAPO, showing that supportive sharing and competing supervision each provide useful learning signals. Full MCPO reaches 55.85, exceeding positive attraction alone by 2.69 points and negative separation alone by 2.18 points, with improvements over both controls in every domain. The gain over negative separation demonstrates the additional value of supportive sharing; the gain over positive attraction demonstrates the additional value of competing supervision. Together, these comparisons support the joint use of both interaction signs: positive attraction reinforces compatible reasoning structures, while negative separation maintains distinctions between competing ones.

Evidence for sharing at the reasoning-segment level. The controls complement the granularity and scope ablations rather than relying on task averages alone. Intra-domain and cross-domain supervision separately reach 53.76 and 53.92, while their combination reaches 55.85, supporting consolidation within domains together with sharing across domains. Panel (b) further shows that the units used for relation construction matter: $K_{\max} = 16$ reaches 55.85, compared with 54.08 at 8 and 55.02 at 64. The performance thus depends on both how reasoning is partitioned and how the resulting relations are used. The evolving signed relations in Figure 3(c) and Appendix D.1, the representation geometry in Appendix D.2, and the Math-to-Science cases in Appendix E.1 provide complementary views of this behavior. Across these results, improvements depend on meaningful pair assignments, suitable segment granularity, and the joint use of supportive and competing examples. This converging evidence supports structured knowledge sharing as an explanation for the gains of MCPO beyond a generic representation loss. The controls establish downstream utility of this structure, and the local relation-preservation conditions are derived separately in Appendix B.4.

D.5 Ablation Diagnostics Support the Structured Contrastive Design

The ablations in Table 2 trace each component of MCPO back to the claim of the introduction that both interaction signs deserve utilization rather than mitigation, and to the formulation in Section 2.3.

Panel (a) isolates cross-domain knowledge sharing and intra-domain consolidation, and Panel (b) varies the maximum number of reasoning segments per response. Panel (c) varies the signed relation-retention ratio q_R, testing the trade-off between discarding weak relations and retaining enough positive and negative evidence for each anchor. Panel (d) provides an additional generality check. Together, these ablations evaluate the separate sharing scopes, segmentation granularity, signed selection rule, and backbone generalization.

D.6 MCPO Loss Weight and Actual Update Contribution

This subsection clarifies how the auxiliary objectives of Section 3.3 actually enter the shared update. The coefficient multiplying the MCPO loss is not itself the fraction of the parameter update contributed by MCPO. Let $\mathcal{L}_{\mathrm{GRPO}}$ denote the prompt-local policy loss and let $\mathcal{L}_{\mathrm{MCPO}}$ denote the weighted sum of the cross-domain and intra-domain contrastive losses. With coefficient λ_{cl}, the gradient entering the common backward pass is

$$g_{\mathrm{tot}} = \nabla_\theta \mathcal{L}_{\mathrm{GRPO}} + \lambda_{\mathrm{cl}} \nabla_\theta \mathcal{L}_{\mathrm{MCPO}}. \tag{58}$$

The coefficient therefore scales the MCPO gradient, but the resulting parameter update also depends on the norm and direction of both gradients, as well as on gradient clipping and the adaptive preconditioning of the optimizer. A loss-value ratio alone cannot determine the update ratio.

For monitoring, we distinguish the scalar-loss ratio from the gradient-level contribution. The former is

$$r_{\mathrm{loss}} = \frac{|\lambda_{\mathrm{cl}} \mathcal{L}_{\mathrm{MCPO}}|}{|\mathcal{L}_{\mathrm{GRPO}}|}, \qquad r_{\mathrm{grad}} = \frac{\|\lambda_{\mathrm{cl}} \nabla_\theta \mathcal{L}_{\mathrm{MCPO}}\|_2}{\|\nabla_\theta \mathcal{L}_{\mathrm{GRPO}}\|_2}. \tag{59}$$

In practice, r_{loss} should use the absolute policy-gradient micro-loss in its denominator because the signed GRPO loss can pass close to zero. The quantity that directly measures the relative force of the auxiliary branch before optimizer preconditioning is r_{grad}; the gradient cosine $\rho = \langle g_{\mathrm{GRPO}}, g_{\mathrm{MCPO}} \rangle / (\|g_{\mathrm{GRPO}}\|_2 \|g_{\mathrm{MCPO}}\|_2)$ additionally indicates whether the two objectives reinforce or oppose one another. When Adam-style moments or gradient clipping are active, the exact parameter displacement should be reported separately as $\|\Delta\theta_{\mathrm{MCPO}}\|_2 / \|\Delta\theta_{\mathrm{tot}}\|_2$.

The current run records a nonzero MCPO branch and a bounded weighted micro-loss ratio throughout training, so the influence of the three coefficients on the actual update is controlled by construction rather than by gradient-magnitude drift. Within one update, the realized contribution is governed by the product structure $\lambda_{\mathrm{cl}}(\lambda_{\mathrm{cross}} \mathcal{L}_{\mathrm{cross}} + \lambda_{\mathrm{intra}} \mathcal{L}_{\mathrm{intra}})$ of Eq. (12), while the cosine-clipped scores and the per-response $GT_{p,g}$ coefficient normalization analyzed in Appendix C.3 keep both the losses and $\|\boldsymbol{h}_{\mathrm{MCPO}}\|_2$ in a bounded range. Changing any coefficient therefore rescales the auxiliary update proportionally, and the stability threshold of Proposition 9 turns the logged margins into an explicit sufficient bound for λ_{cl}. The ratios r_{loss}, r_{grad}, and ρ defined above are the loss-level and gradient-level views of the same contribution. Concretely, the default $\lambda_{\mathrm{cl}} = 0.05$ keeps the weighted MCPO micro-loss at about 9.6% of the absolute policy-gradient micro-loss across the completed 500-step run, and the batch-level monitor $\boldsymbol{g}_{\mathcal{B}}^{\top} \boldsymbol{h}_{\mathrm{MCPO}}$ of Proposition 8 tracks how strongly the auxiliary direction aligns with the GRPO update.

D.7 Computational Profiling Protocol

We distinguish an implementation profile from a matched overhead comparison. The implementation profile uses the final 500-step Qwen3-4B run on the same eight-H200 node described above. Profiling synchronizes CUDA immediately before and after token-sketch construction and the subsequent segmentation–relation–loss region. The latter region includes candidate retrieval, signed pair scoring, construction of both contrastive losses, FIFO bank update, enabled qualitative case logging, and metric construction. These timers end before the joint backward call.

VERL records the mean component time across actor micro-batches. One update contains 128 prompts with eight rollouts each. After data-parallel partitioning over eight GPUs and a micro-batch size of two responses per GPU, each worker executes 64 serial micro-batches. We multiply the logged per-micro-batch mean by 64 when comparing the internal branch with the complete actor-update and training-step timers. Throughput is the framework metric in processed tokens per second per GPU, and peak memory is the maximum memory allocated by CUDA. The sketch-graph measurement is the change in allocated CUDA memory across sketch construction; bank memory counts the valid

Table 7: Computation profile over 500 updates of the final Qwen3-4B MCPO run. Percentiles are taken across the update-level logged values. Memory entries are allocated-memory measurements or explicitly identified payload estimates, not counterfactual overhead relative to GRPO.

Quantity	Unit	Mean	95th pct.	Maximum
Sketch construction per micro-batch	ms	16.14	19.36	32.40
Segmentation, relations, and loss per micro-batch	ms	107.98	126.97	143.39
Total MCPO branch per micro-batch	ms	124.12	143.45	161.57
Accumulated MCPO branch per update	s	7.94	9.18	10.34
MCPO branch / actor update	%	8.296	9.428	10.152
MCPO branch / complete training step	%	2.420	2.785	3.028
Incremental sketch-graph allocation	GiB	3.73	4.43	5.16
Valid source-sketch payload in bank	MiB	0.592	0.622	0.625
Realized pair-score arithmetic	MFLOPs	0.718	0.856	0.983
Complete training-step time	s	328.04	342.47	370.14
Throughput	tokens/s/GPU	1,427.12	1,609.32	1,725.85
Peak allocated GPU memory	GiB	135.78	136.02	136.02

FP16 sketch payload and excludes small record metadata. Appendix D.8 reports all 500 updates, together with the analytical bounds and the generated figure.

D.8 Computational Complexity and Practical Overhead

This subsection quantifies the practical cost of the mechanisms introduced in Section 3, so that the contrastive objectives remain deployable at the scale of the main runs. We follow the component-wise reporting used in scalable multi-domain policy optimization (Liang et al., 2026). The analysis below first identifies the extra tensor operations and storage introduced by MCPO, then reports the instrumented cost of the final 500-step implementation. Analytical work, internal synchronized timers, and matched end-to-end overhead are kept as separate quantities.

Notation and implementation. Let B be the number of responses in one actor micro-batch, S the complete non-padding sequence length, T the valid response-prediction length, V the vocabulary size, and H the final hidden width. Each response contains at most $K_{\max}$ segments, where $K_{\max}$ is the segment budget, and retrieves at most C source responses in each cross-domain and intra-domain scope. Let k denote the number of retained categorical coordinates. A local FIFO bank stores at most M responses for each of the D domains.

MCPO introduces neither an additional rollout nor a second actor or teacher forward pass. The actor forward already produces the logits and sampled-token log probabilities; a hook captures the corresponding final hidden factors. The sampled-token log probability recovers the softmax normalizer used by the sparse categorical residual, so the sketch branch does not evaluate a second full-vocabulary log-sum-exp. Predictive entropy over the non-padding sequence requires $\mathcal{O}(BSV)$ work. The analytical configuration uses $K_{\max} = 16$ and no entropy regularization. Predictive entropy is computed to select segmentation boundaries and contributes no regularization term to the policy objective. We include its computation in the MCPO cost.

Why compact segment gradients are required. Materializing one exact dense gradient for every segment is not practical. If a model has P parameters, storing FP32 full-model gradients for $BK_{\max}$ segments requires $4BK_{\max}P$ bytes before accounting for autograd state. For Qwen3-4B, $P \approx 4 \times 10^9$, $B = 2$, and $K_{\max} = 16$, this is approximately 512 GB (477 GiB) in aggregate. Even restricting the exact gradient to the language model head would store $BK_{\max}VH$ FP32 values. With $V = 151{,}936$ and $H = 2{,}560$, this amounts to 46.4 GiB per actor micro-batch. MCPO instead uses the compact factorized output-gradient representation analyzed in Appendix B.4.

Segmentation and feature construction. Response-relative quantiles and ordered cutpoint resolution require $\mathcal{O}(BT \log T)$ work under a conservative sorting bound, followed by $\mathcal{O}(BT)$ indexing. For valid response positions, vocabulary top-k selection scans BTV logits, sparse residual construction processes BTk entries, and the full hidden CountSketch processes BTH coordinates. For a fixed compact mapping, feature construction and segment aggregation add work linear in the number

of processed response positions. Thus,

$$\mathcal{C}_{\text{construct}} = \mathcal{O}(BT \log T + BTV + BTH + BTk)\,. \tag{60}$$

Remove-padding restricts top-k selection and sketch construction to valid response-prediction positions; prompt tokens are not processed by this branch.

Candidate retrieval and relation scoring. Before the current records are enqueued, one target can inspect at most $Q = B + DM$ current and bank records. The implemented filtering and recency sort therefore have the conservative CPU-side bound $\mathcal{O}(BQ \log Q)$. Candidate capping then limits each scope to C source responses. Since a target and source response each contain at most $K_{\max}$ segments, the number of evaluated segments and their scoring cost under the fixed representation satisfy

$$N_{\text{pair}} \le 2BCK_{\max}^2, \qquad \mathcal{C}_{\text{pair}} = \mathcal{O}(BCK_{\max}^2). \tag{61}$$

The factor two accounts for the cross-domain and intra-domain retrieval scopes. Sign masks and the two multi-positive log-sum-exp losses are linear in N_{pair}. Holding the compact mapping fixed and combining the components gives

$$\begin{aligned}\mathcal{C}_{\text{MCPO}} = \mathcal{O}\big(&BSV + BT \log T + BTV + BTH + BTk \\ &+ B(B + DM) \log(B + DM) + BCK_{\max}^2\big).\end{aligned} \tag{62}$$

The BSV term accounts for the entropy computation required by segmentation in this configuration. Candidate capping makes the GPU relation-scoring term independent of D once C is reached; bank storage and the current CPU retrieval scan still grow linearly with the number of domains.

Explicit storage and measured graph allocation. Differentiable target token sketches use FP32, whereas stopped source-segment sketches in the FIFO bank use FP16. For the fixed representation, feature storage scales with token and segment counts, while the FP32 pair-score payload occupies four bytes per pair:

$$\mathcal{M}_{\text{features}} = \mathcal{O}(BT + DMK_{\max}), \qquad \mathcal{M}_{\text{pair,payload}} \le 4N_{\text{pair}} \le 8BCK_{\max}^2 \quad \text{bytes.} \tag{63}$$

The pair-score expression is a numeric-payload bound, not a peak-memory bound. Top-k values and indices, FFT workspaces, saved tensors, and autograd metadata are transient and can be substantially larger. We therefore instrument the increment in allocated CUDA memory across differentiable token-sketch construction. The bank is GPU-local and invokes no MCPO-specific all-gather or all-reduce. This avoids additional communication, but also means that the source pool of a worker covers only examples observed locally.

Default analytical scale. The default analytical configuration uses $B = 2$, $T = 8{,}192$, $V = 151{,}936$, $H = 2{,}560$, $K_{\max} = 16$, $C = 16$, $D = 5$, $M = 32$, and $k = 32$. At maximum shape, response top-k scans 2.49 billion logit entries and the hidden CountSketch processes 41.9 million coordinates. Relation scoring evaluates at most $16{,}384$ segment pairs. The FP32 pair-score payload is at most 64.0 KiB per GPU; feature-construction latency and memory allocation are measured below. These are upper shape counts; valid-token masking and shorter responses reduce realized work. We do not equate the scanned-entry count with FLOPs because the top-k wall time depends on the GPU kernel and memory layout.

Profiling setup. We profile the completed 500-step Qwen3-4B run on one node with eight NVIDIA H200 GPUs. CUDA synchronization brackets sketch construction and the subsequent segmentation–relation–loss region. VERL averages these timers over the local actor micro-batches before logging one value per update. With 128 prompts, eight rollouts per prompt, eight data-parallel workers, two responses per actor micro-batch, and one PPO epoch, each worker executes

$$N_\mu = \frac{128 \times 8}{8 \times 2} \times 1 = 64 \tag{64}$$

serial actor micro-batches per update. We therefore estimate the accumulated pre-backward MCPO branch time as $T_{\text{branch}} = N_\mu\, \bar{t}_{\text{MCPO}}$, where $\bar{t}_{\text{MCPO}}$ is the logged per-micro-batch mean. The second timer includes segmentation, candidate retrieval, relation scoring, contrastive-loss construction, bank update, enabled case logging, and metric construction. It ends before the common backward call, so it does not isolate the extra backward work induced by the differentiable target sketches. Panels (b) and (c) of Figure 7 report the per-micro-batch latency and the memory and throughput traces over all 500 updates, and Table 7 lists the corresponding update-level statistics.

Interpretation and reporting boundary. Sketch construction takes 16.14 ms per actor micro-batch on average, while the remaining MCPO branch takes 107.98 ms. After accounting for all 64 local micro-batches, the instrumented pre-backward branch occupies 8.296% of actor update time and 2.420% of the complete training step. The summed logged training-step time over 500 updates is 45.56 hours, of which the accumulated instrumented branch accounts for 1.10 hours. Rollout generation averages 209.83 seconds per step and the complete actor update averages 95.74 seconds, so these remain the dominant components. The realized pair-score arithmetic is below one MFLOP on average, showing that relation inner products are not the principal cost; retained differentiable sketch graphs and the surrounding relation pipeline dominate the auxiliary branch. The FIFO bank itself is small, averaging 0.592 MiB of valid FP16 segment-sketch payload.

E Case Study

E.1 Case Study: Math-to-Science Knowledge Sharing on SciBench Questions

The main results and representation analysis indicate that MCPO benefits multi-domain RL by converting compatible cross-domain interactions into knowledge sharing. We further examine this mechanism through Science questions that require mathematical reasoning, where transferable Math solution structures can directly support quantitative Science problem solving. This case study is therefore intended as qualitative evidence for cross-domain knowledge sharing through mutually augmented examples.

Evaluation protocol. SciBench (Wang et al., 2024) is well suited for examining cross-domain knowledge sharing because its questions often combine science-specific constraints with formal mathematical reasoning, such as equation manipulation, numerical computation, unit conversion, and quantitative physical inference. Solving these items requires applying transferable Math reasoning structures to Science problem solving while preserving the domain-specific semantics of the question. We therefore compare DAPO and MCPO-DAPO on the same SciBench evaluation split with identical decoding settings, and report two representative paired cases with model responses and mechanism-oriented analysis. Red boxes show the DAPO responses, green boxes show the MCPO-DAPO responses, and dark red text marks the key behavioral differences. For readability, the boxed responses preserve the original wording of key reasoning steps, while ellipses (. . .) omit repetitive or semantically unclear passages without substantive reasoning progress.

Case 1: Hooke-law work calculation. **Question ID:** Science_SciBench_thermo_14

Question. A muscle fiber contracts by $3.5\,\mathrm{cm}$ and in doing so lifts a weight. Calculate the work performed by the fiber. Assume the muscle fiber obeys Hooke's law $F = -kx$ with a force constant $k = 750\,\mathrm{N\,m^{-1}}$. Remember to round the answer to two numbers of decimal places.

Ground truth: 0.46

DAPO Response — Prompt-local RL baseline

<think>
Okay, let's see. The problem is about calculating the work performed by a muscle fiber that contracts by $3.5\,\mathrm{cm}$ and lifts a weight. The muscle fiber obeys Hooke's law with a force constant of $750\,\mathrm{N/m}$. . . . First, I need to recall how work is calculated when a force is varying. Since the muscle fiber follows Hooke's law, the force isn't constant but depends on the displacement. Hooke's law is $F = -kx$, where k is the spring constant and x is the displacement. The negative sign indicates that the force is restorative, but since we're interested in the work done by the muscle fiber, maybe I can ignore the direction and focus on the magnitude.
Work done by a variable force is the integral of $F\,dx$ over the distance. So, $W = \int F\,dx$ from 0 to x. . . . So, the work done by the muscle fiber would be the integral of $F\,dx$ from $x = 0$ to $x = 3.5\,\mathrm{cm}$. But since F is $-kx$, then $W = \int(-kx)\,dx$ from 0 to $0.035\,\mathrm{m}$. Wait, but is that correct? Wait, when the muscle contracts, the force is acting in the opposite direction to the displacement. But when calculating work, it's the dot product of force and displacement. So, if the force is in the opposite direction to the displacement, the work done by the force would be negative. But the problem says the muscle fiber lifts a weight. So maybe the work done by the muscle fiber is positive? I need to be careful here.

But let's stick to the physics. The work done by the muscle fiber is the integral of $F\,dx$. Since $F = -kx$, then $W = \int(-kx)\,dx$ from 0 to x. The integral of $-kx\,dx$ is $-0.5kx^2$. ... But this would be negative. However, since the muscle is lifting a weight, maybe the work done by the muscle is positive. Wait, maybe I have the direction wrong. ... But this is confusing. Let me just use the formula as given in the problem. Since the force is $F = -kx$, the work is

$$W = \int_0^{0.035} -kx\,dx = -\frac{1}{2}k(0.035)^2.$$

Plugging in $k = 750\,\mathrm{N/m}$, I get

$$W = -\frac{1}{2}\cdot 750\cdot(0.035)^2 = -0.459375\,\mathrm{J}.$$

So, rounding to two decimal places, the work is about $-0.46\,\mathrm{J}$.
</think>
<answer>

$$\boxed{-0.46}$$

</answer>

Analysis. DAPO reaches the transferable calculus template $W = \int F\,dx$, but it does not bind this Math structure to the Science meaning of "work performed" by a lifting muscle. The rollout repeatedly reopens the sign convention and finally keeps the restoring-force sign from Hooke's law, turning a compatible Math-to-Science interaction into a negative-work answer.

MCPO-DAPO Response — Structured knowledge sharing

<think>
Okay, let's see. The problem is about calculating the work performed by a muscle fiber that contracts by $3.5\,\mathrm{cm}$ and lifts a weight. They mentioned Hooke's law, $F = -kx$, with k being $750\,\mathrm{N/m}$. So, first, I need to recall how work is calculated when a force is applied over a distance. Work is usually force multiplied by distance, but since the force here isn't constant—it's changing according to Hooke's law—it's probably a variable force. So, the work done should be the integral of the force over the distance.
Hooke's law says the force is proportional to the displacement x, so $F = -kx$. But since we're dealing with magnitude here, maybe I can ignore the negative sign because work is a scalar quantity. So, the force exerted by the muscle fiber as it contracts is $F = kx$. Wait, but when the muscle contracts, it's pulling against the weight. So, the force is opposing the displacement? Hmm, maybe the negative sign indicates direction, but for work, we need to consider the direction of the force relative to the displacement. Since the muscle is contracting and lifting the weight, the force is in the same direction as the displacement. So, maybe the negative sign can be ignored here because we're considering the magnitude.
So, the work done should be the integral from 0 to x of $F\,dx$. So, integrating $kx\,dx$ from 0 to $0.035\,\mathrm{m}$. Because $3.5\,\mathrm{cm}$ is $0.035\,\mathrm{m}$. The integral of $kx\,dx$ is $(1/2)kx^2$. So, plugging in the numbers, k is $750\,\mathrm{N/m}$, x is $0.035\,\mathrm{m}$. So, $(1/2)\cdot 750\cdot(0.035)^2$. ... Then multiply by $1/2$: $0.91875/2 = 0.459375\,\mathrm{J}$. Yes, that's correct. So the work performed is $0.459375\,\mathrm{J}$. Rounded to two numbers of decimal places, So $0.46\,\mathrm{J}$. ... The force constant is in $\mathrm{N/m}$, so units are consistent. So the work should be in Joules. So the final answer is $0.46\,\mathrm{J}$.
</think>
<answer>

$$\boxed{0.46}$$

</answer>

Analysis. MCPO-DAPO uses the same variable-force integral, but anchors it to the Science context by interpreting the Hooke-law sign as direction and the requested quantity as work magnitude. The supportive signal of the mutually augmented example therefore keeps the transferred Math structure aligned with the physical semantics of lifting, yielding the correct positive work.

Case 2: Virial theorem for a quartic potential. **Question ID:** Science_SciBench_quan_30

Question. A particle is subject to the potential energy $V = ax^4 + by^4 + cz^4$. If its ground-state energy is 10 eV, calculate $\langle V \rangle$ for the ground state. Remember to round the answer to two numbers of decimal places.

Ground truth: 3.33

DAPO Response — Prompt-local RL baseline

<think>
Okay, let's see. The problem is about a particle in a potential energy field given by $V = ax^4 + by^4 + cz^4$. The ground-state energy is given as 10 eV, and we need to find the expectation value of V for the ground state. ... For a particle in a potential, the total energy is the sum of the kinetic energy and the potential energy. So, the ground-state energy is the sum of the ground-state kinetic energy and the ground-state potential energy. But wait, the question is asking for the expectation value of the potential energy, $\langle V \rangle$. So, if the total ground-state energy is 10 eV, then maybe $\langle V \rangle$ is part of that total energy. But how do I find $\langle V \rangle$ specifically?
Hmm. I remember that for certain potentials, like the harmonic oscillator, the expectation value of the potential energy is equal to the expectation value of the kinetic energy. But this potential is different because it's a quartic potential (x^4, y^4, z^4). So, maybe that property doesn't hold here. ... But the problem doesn't specify any particular relations between a, b, c. However, the ground-state energy is given as 10 eV. But how can we find $\langle V \rangle$ without knowing a, b, c? ... The ground-state energy is the total energy, which is the sum of kinetic and potential energies.
Wait, maybe the question is referring to the fact that for a particle in a potential, the ground-state energy is the expectation value of the Hamiltonian. The Hamiltonian is H = kinetic energy + potential energy. So, the ground-state energy is $\langle \psi | H | \psi \rangle = \langle T \rangle + \langle V \rangle$. So, if the total energy is 10 eV, then $\langle T \rangle + \langle V \rangle = 10$ eV. But the question is asking for $\langle V \rangle$. But without knowing $\langle T \rangle$, how can we find $\langle V \rangle$? ... But since the problem doesn't give any specific information about the kinetic energy or a relation between $\langle T \rangle$ and $\langle V \rangle$, I cannot determine $\langle V \rangle$ from the given ground-state energy alone. So, I cannot give a numerical answer without additional information. ... (The response repeatedly revisits the missing relation between $\langle T \rangle$ and $\langle V \rangle$ until the rollout is truncated.)
</think>
<answer>

$$\boxed{null}$$

</answer>

Analysis. DAPO recognizes that total energy decomposes into kinetic and potential parts, but it cannot import the power-law relation needed for a quartic potential. The Science question is therefore treated as missing coefficients, instead of as an instance where a transferable homogeneous-potential structure can determine $\langle V \rangle$.

MCPO-DAPO Response — Structured knowledge sharing

<think>
Okay, so I need to find the expectation value of the potential energy $\langle V \rangle$ for the ground state of a particle in the given potential $V = ax^4 + by^4 + cz^4$. The ground-state energy is given as 10 eV. ... The potential energy here is a sum of terms involving x^4, y^4, and z^4. Since the potential is separable into three independent terms, maybe the ground state wavefunction can be written as a product of three separate wavefunctions for each coordinate. If that's the case, then the expectation value of V would be the sum of the expectation values of each term individually. So $\langle V \rangle = \langle ax^4 \rangle + \langle by^4 \rangle + \langle cz^4 \rangle$.
But wait, the ground-state energy is given as 10 eV. How is that related to the expectation value of the potential energy? ... The total energy $E = \langle T + V \rangle$. But the problem states that the ground-state energy is 10 eV. So, $E = \langle T + V \rangle = 10$ eV. But how do I find $\langle V \rangle$? Wait, I need to consider the virial theorem. For a potential that is a power law, the virial theorem relates the expectation values of kinetic and potential energy. ... For example, for a potential $V(r) \propto r^n$, the virial theorem states that $2\langle T \rangle = n\langle V \rangle$. But in this case, the potential is a sum of terms each involving different coordinates. However, each term is a quartic term.
If the potential is $V = ax^4 + by^4 + cz^4$, then each term is a quartic in each coordinate. ... For a potential $V(x) = kx^n$, the virial theorem states that $2\langle T \rangle = n\langle V \rangle$. So, if each direction has a potential term like kx^4, then for each direction, $2\langle T_i \rangle = 4\langle V_i \rangle$. Therefore, for each direction, the kinetic energy is $2\langle V_i \rangle$. ... Therefore, the total kinetic energy is $2(\langle V_x \rangle + \langle V_y \rangle + \langle V_z \rangle)$. Therefore, the total energy

$$E = \langle T_{\text{total}} + V_{\text{total}} \rangle = 2(\langle V_x \rangle + \langle V_y \rangle + \langle V_z \rangle) + (\langle V_x \rangle + \langle V_y \rangle + \langle V_z \rangle) = 3\langle V_{\text{total}} \rangle.$$

Therefore, $\langle V_{\text{total}} \rangle = E/3$. Since the ground-state energy E is $10\,\text{eV}$, then $\langle V \rangle = 10\,\text{eV}/3 \approx 3.333\,\text{eV}$. So, the expectation value of V is 3.33

<answer>

$$\boxed{3.33}$$

</answer>

Analysis. MCPO-DAPO first recasts the Science potential as a homogeneous quartic structure and then transfers the virial relation for power-law potentials. This supportive relation supplies the missing link between $\langle T \rangle$ and $\langle V \rangle$, converting $E = 10\,\text{eV}$ into $\langle V \rangle = E/3$.

E.2 Mechanism analysis

Together, these cases show how MCPO changes the role of Math domain structures in Science rollouts. DAPO often recalls local formulas but leaves them isolated from the surrounding Science semantics, whereas MCPO-DAPO binds compatible reasoning structures to the target context and reaches verifier-aligned answers. In the terminology of the method, the gradient variation test admits these transferable Math structures as positive members of mutually augmented examples, so cross-domain knowledge sharing strengthens them as supportive signals instead of leaving them to compete with domain-specific conventions. This qualitative evidence complements the aggregate Science results by showing how cross-domain knowledge sharing changes the reasoning structures at the rollout level.

F Discussions and Limitations

F.1 Applicability of MCPO to Multi-Domain Pre-Training

MCPO is developed for on-policy multi-domain RL, but its central construction is not tied to a particular verifier. It requires an additive sequence objective, a partition of that objective into local reasoning units, and gradients whose pairwise interactions can be evaluated before response-level aggregation. These requirements also hold for next-token prediction on a heterogeneous pre-training corpus: the token cross-entropy can be regrouped over contiguous segments, and the signed interaction between two segment gradients still predicts the first-order change in one segment loss after learning from the other.

A direct pre-training instantiation nevertheless differs from the method studied in this paper. GRPO advantages and clipping determine the token coefficients in our segment gradients, whereas supervised pre-training would use the coefficients of its token likelihood objective. On-policy rollouts would also be replaced by observed document continuations, and domain identifiers would need to be supplied or inferred. Entropy-based boundaries, the signed relation score, and the two contrastive scopes can be retained after these substitutions. We therefore view pre-training as a structurally compatible extension, not as an empirical claim established by the present experiments.

F.2 Non-Uniform Domain Importance in Multi-Domain Training

Equation (1) already permits non-uniform domain weights α_d. In practice, importance can enter MCPO at two distinct locations. First, the target sampling distribution determines how frequently a domain contributes prompt-local GRPO gradients and target reasoning segments. Second, relation weights can determine how strongly a target domain receives cross-domain sharing and intra-domain consolidation. These controls should not be conflated: oversampling a target domain changes both its ordinary RL signal and the number of available relations, whereas changing its contrastive weight modifies only the sharing term.

Let $d(u)$ denote the target domain of segment u. A weighted extension of the implemented objective can be written as

$$\mathcal{L}_{\text{total}}^{\text{weighted}} = \mathcal{L}_{\text{GRPO}} + \lambda_{\text{cl}} \sum_{d=1}^{D} \omega_d \left(\lambda_{\text{cross}} \mathcal{L}_{\text{cross}}^{(d)} + \lambda_{\text{intra}} \mathcal{L}_{\text{intra}}^{(d)} \right), \qquad \sum_{d=1}^{D} \omega_d = 1, \tag{65}$$

where each scope loss is averaged over active targets with $d(u) = d$. The coefficients ω_d can encode application priorities without changing the signed relation rule. The current five-domain experiments use the common target mixture and fixed scope weights in Table 4; adaptive or externally specified domain priorities remain to be evaluated.

F.3 Why Joint Learning Cannot Reproduce the Cross-Domain Mechanism

F.3.1 Prompt Aggregation Hides Local Signed Interactions

Joint GRPO does permit domains to interact because all prompts update the same parameters. Its cross-prompt effect, however, is available to the optimizer only through the aggregate gradient. Expanding one prompt-pair interaction over its segment pairs (Appendix B.1) shows that it is a sum of signed terms, which aggregation reduces to a positive-minus-negative remainder. Positive and negative terms can therefore cancel before the optimizer receives any indication of which local reasoning decisions were supportive and which were competing. Changing the batch mixture or the prompt weight rescales this sum; it does not recover the terms that were lost inside it.

F.3.2 Segment-Wise Assignment Preserves Support and Competition

MCPO evaluates $\kappa_{u \leftarrow v}$ before this aggregation. Each target segment consequently receives a pair-indexed relation set, separated by cross-domain and intra-domain scope. Positive scores identify positive members of the mutually augmented examples, and negative scores identify negative members that participate in competition. The source representation and relation membership are stopped, while the target score remains differentiable. Hence the contrastive gradient is assigned to the target reasoning segment that exhibits the measured relation, rather than to an undifferentiated prompt total.

F.3.3 Implications for Knowledge Sharing

This distinction concerns the objective, rather than merely the order of the training data. Joint training optimizes the sum of prompt-local GRPO losses. MCPO additionally optimizes the relative ordering of supportive and competing local updates in Eqs. (10) and (11). It can therefore strengthen a supportive segment relation even when that relation is masked by other opposed terms in the same pair of response gradients. Conversely, it does not assert that every cross-domain example is helpful: the sign test determines whether a candidate enters the mutually augmented set as a positive or a negative member. This selective local credit assignment is the mechanism that ordinary joint learning does not provide. Furthermore, we view self-supervised contrastive learning as a promising complement to trial-and-error on-policy post-training of large reasoning models, particularly for multi-domain training, continual learning, and model evolution.

F.4 Limitations

First, MCPO represents cross-domain interactions with reasoning segments selected by response-relative entropy quantiles. This proxy may not coincide with the true interaction strength between reasoning structures, so the resulting contrastive signals can be incomplete. Related analyses of entropy-based segmentation (Ai et al., 2026; Guo et al., 2025b; Cui et al., 2025) and our own diagnostics nevertheless indicate that high-entropy segments correspond to the reasoning structures that carry the most important reasoning information of each rollout, which supports using them as the basic units of cross-domain interaction.

Second, the cross-batch source bank in Appendix D.3 widens the candidate pool, yet the bank and freshness analysis shows that even with a fresh bank, knowledge sharing still does not span batches that are far apart in the training stream, so some prompts produce rollouts that can not be explicitly shared before their records expire. However, MCPO is by nature cross-domain transfer, and the experiments have repeatedly shown that the knowledge-sharing signal has an upper bound that is not determined by hyperparameters. The unshared distant batches therefore expose another defect of multi-domain post-training frameworks rather than a defect of the cross-domain knowledge sharing that MCPO resolves, although we consider the two problems to be partially homologous. In future work, we plan to start from the base-model architecture and instantiate the bank inside concrete parameter layers of the model, such as the KV cache or the mixture-of-experts(MOE) routing of the transformer, so that knowledge sharing connects training with the model itself.