Co-Evolving Policy Distillation

Naibin Gu[1,2]*, **Chenxu Yang**[1,2]*, **Qingyi Si**[3]*, **Chuanyu Qin**[1,2], **Dingyu Yao**[1,2], **Peng Fu**[1,2]† , **Zheng Lin**[1,2], **Weiping Wang**[1], **Nan Duan**[3], **Jiaqi Wang**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]JD.com
{gunaibin,fupeng}@iie.ac.cn, siqingyi.phoebus@jd.com

Abstract

RLVR and OPD have become standard paradigms for post-training. We provide a unified analysis of these two paradigms in consolidating multiple expert capabilities into a single model, identifying capability loss in different ways: mixed RLVR suffers from inter-capability divergence cost, while the pipeline of first training experts and then performing OPD, though avoiding divergence, fails to fully absorb teacher capabilities due to large behavioral pattern gaps between teacher and student. We propose **Co-Evolving Policy Distillation** (**CoPD**), which encourages parallel training of experts and introduces OPD during each expert's ongoing RLVR training rather than after complete expert training, with experts serving as mutual teachers (making OPD bidirectional) to co-evolve. This enables more consistent behavioral patterns among experts while maintaining sufficient complementary knowledge throughout. Experiments validate that CoPD achieves all-in-one integration of text, image, and video reasoning capabilities, outperforming strong baselines such as mixed RLVR and MOPD, and even surpassing domain-specific experts[1].

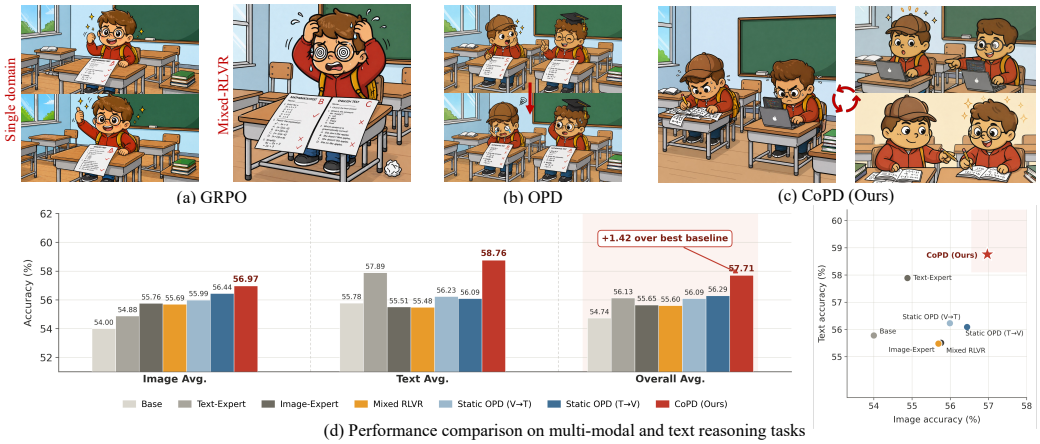

Figure 1: Illustration of the training paradigms compared in this work. (a) Mixed-data RLVR trains one policy on multiple domains, causing capability interference. (b) Static OPD performs distillation only after independent specialization, whereas (c) CoPD addresses the limitations of mixed-data RLVR (a) and static OPD (b) by letting two branches co-evolve as teachers and students across domains, achieving the best performance.

* denotes equal contribution. † denotes corresponding author.
[1]The code is available at `https://github.com/gccnlp/CoPD`.

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

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) [1–4] has emerged as the dominant post-training paradigm for large models, driving rapid progress on visual reasoning [5, 6], text reasoning [7, 8], video understanding [9, 10], and other domains [11, 12]. However, training a single model on mixed-capability data with RLVR often exhibits a capability trade-off [13, 14], where gains in one capability come at the expense of another, as illustrated in Figure 1(a). We refer to this phenomenon as *capability divergence*: distinct capabilities favor different optimization directions, making it difficult for a single training run to advance all of them simultaneously. Since no single RLVR run can master all capabilities at once, the field has converged on a two-stage pipeline: separate copies of the base model are first trained with RLVR on each capability to obtain specialized experts, and these experts are then consolidated into a unified policy model through on-policy distillation (OPD) [15], also known as Multi-teacher OPD (MOPD) when multiple experts serve as teachers simultaneously [13, 16]. By separating capability-specific training from cross-capability consolidation, OPD avoids the gradient conflicts caused by capability divergence , which has made OPD a widely adopted approach for capability consolidation.

However, the way OPD is currently applied leaves an important issue unaddressed. In current practice, each expert is first trained until convergence on its own capability data, and then used as a fixed teacher for distillation into a separate student model, as illustrated in Figure 1(b). We argue that by the time distillation begins, the teacher has drifted too far from the student for its supervision to be effectively absorbed. To examine this, we measure the behavioral distance between teacher and student via *top-k token overlap* along on-policy trajectories. Our pilot study (§3) shows that effective distillation requires teacher and student to remain behaviorally close. When they are too far apart, the teacher's supervision becomes difficult for the student to absorb, which is precisely where the static OPD pipeline falls short. We therefore argue that teacher and student should *co-evolve*, with their behavioral distance kept within the absorbable range throughout training, rather than being set up as a static expert and a passive learner only at the moment of distillation.

Building on this insight, we propose Co-Evolving Policy Distillation (CoPD), which unifies capability exploration and consolidation into a single co-evolving process. CoPD maintains parallel training branches, each associated with a different capability. Instead of freezing each branch as a static teacher, CoPD interleaves branch-specific RLVR with cross-branch mutual distillation throughout training, so that each branch continuously deepens its own expertise while serving as an evolving teacher for the other. Concretely, training proceeds in alternating phases, as illustrated in Figure 1(c). In the *RLVR phase*, each branch independently optimizes on its own capability data, deepening branch-specific expertise and *opening up* the behavioral distance between branches, so that subsequent distillation has substantive new knowledge to transfer. In the *mutual OPD phase*, each branch generates rollouts on the other branch's data and receives token-level supervision from the other, *closing* the behavioral distance and absorbing what the other branch has newly learned with high efficiency, since the two branches remain within each other's effective range. The two phases are tightly complementary: RLVR drives the branches apart enough to make distillation informative, while interleaving keeps them close enough for distillation to be absorbable. Mutual OPD then exploits this window to transfer what each branch has learned to the other. Because all branches start from a shared base and remain tightly coupled through continuous mutual distillation, their parameters do not diverge drastically, and the final unified model can be obtained through simple parameter merging.

We evaluate CoPD in both two-branch and three-branch post-training settings. The two-branch setting unifies text reasoning and multimodal reasoning, with each branch specializing in one capability and serving as an evolving teacher for the other. To demonstrate that CoPD scales beyond the pairwise case, we further evaluate a three-branch setting that jointly consolidates text reasoning, image-text reasoning, and video understanding. CoPD consistently outperforms single-domain expert, mixed-data RLVR and static (M)OPD pipelines in both the two-branch and three-branch settings. Ablation studies confirm the necessity of bidirectional distillation over unidirectional alternatives and the advantage of continuous co-evolution over one-shot distillation.

Our contributions are as follows:

- We identify a key limitation of the prevailing RLVR-then-OPD pipeline through a pilot study showing that OPD gain is inversely related to teacher–student behavioral distance, indicating that distillation suffers when the two drift too far apart. We accordingly argue

that teacher and student should *co-evolve* via cross-branch mutual distillation, a principle that naturally extends to multi-expert parallel training.

- We propose CoPD, which interleaves branch-specific RLVR with cross-branch mutual OPD throughout training. RLVR drives branch-specific exploration, while mutual OPD continuously transfers cross-domain knowledge. This design supplies sufficient new knowledge while keeping it absorbable at every step, achieving all-in-one consolidation without a separate distillation stage.
- Experiments show that CoPD consistently outperforms mixed RLVR, original OPD, MOPD baselines across text, image, video benchmarks. Notably, CoPD surpasses domain-specific experts on both image and text reasoning, turning cross-domain trade-offs into mutual gains.

2 Related Work

RLVR for Large Models. Reinforcement learning with verifiable rewards (RLVR) has become the dominant paradigm for eliciting reasoning capabilities in large models, replacing human-preference reward models with rule-based verifiable reward functions [7, 17–19]. GRPO [1] eliminates the need for a separate value network by estimating advantages from group-level reward statistics, making RLVR training feasible at scale, with subsequent work further improving training stability and data curation [3, 4]. Recently, RLVR has been widely applied to enhance a variety of capabilities, including image reasoning [5, 6], video reasoning [9, 10], and coding and agentic tasks [11, 12]. This work explores how to accommodate multiple capabilities simultaneously within the RLVR framework.

On-Policy Distillation. On-policy distillation (OPD) provides teacher supervision on the student's own generated trajectories, mitigating the train-inference distribution mismatch in off-policy methods [20–22]. By offering dense token-level supervisory signals, OPD is widely used to rapidly improve a target model's capabilities with the guidance of a stronger model [15, 8, 23–25]. Recently, Multi-teacher OPD (MOPD) has been widely adopted in base model post-training, where multiple specialized experts are first trained on different capabilities and then distilled into a single policy model [13, 16]. Through a pilot study (§3), we find that this paradigm suffers from a behavioral mismatch between teacher and student that prevents the student from fully absorbing the teacher's capabilities. Our use of top-k token overlap as a behavioral indicator is inspired by Li et al. [26], while our pilot study takes a complementary perspective by varying the student's sampling temperature to construct teacher–student pairs at controlled overlap levels. Motivated by this finding, CoPD performs mutual distillation *during* expert training rather than after it, keeping teacher and student behaviorally close while preserving the complementary knowledge gap for effective transfer.

3 Unleash the Potential of RLVR and OPD

This section motivates CoPD through three steps. We first formalize the structural costs incurred by mixed-data RLVR and the static OPD pipeline under a unified framework (§3.1). We then hypothesize that effective OPD requires teacher and student to share similar behavioral patterns, and use top-k token overlap as a measurable indicator (§3.2). Finally, we verify both the indicator and its implications through a pilot study, which directly motivates CoPD (§3.3).

3.1 A Unified View of Existing Paradigms

We consider consolidating two capabilities, represented by datasets D_1 and D_2, into a single unified policy. Let $X(D_1, D_2)$ denote the total optimization signal contained in the two datasets, that is, the capability gain that would be realized if the signal from each dataset could be perfectly applied to the unified policy. We analyze each paradigm by how much of X it actually delivers, and what additional cost it incurs. The resulting utility takes the schematic form

$$U_{\mathcal{P}} \;\approx\; a_{\mathcal{P}} \cdot X(D_1, D_2) \;+\; b_{\mathcal{P}}, \tag{1}$$

where $a_{\mathcal{P}} \in [0, 1]$ measures how effectively paradigm $\mathcal{P}$ converts X into actual capability, and $b_{\mathcal{P}} \leq 0$ captures any additional loss beyond imperfect conversion.

Mixed-data RLVR: capability divergence as conflict. When a single model is jointly optimized on $D_1 \cup D_2$, the per-step update averages the gradients induced by the two datasets. Because distinct

capabilities favor different optimization directions, their gradients disagree on capability-specific dimensions, and the disagreement manifests as gradient conflict within each parameter update. The resulting utility takes the form

$$U_{\text{mix}} \;\approx\; X(D_1, D_2) \;-\; \Phi(D_1, D_2), \tag{2}$$

where $\Phi(D_1, D_2) > 0$ is the *capability divergence cost*. Mixed-data RLVR pays this cost as the seesaw effect: gains on one capability are partially canceled by interference from the other. This corresponds to $a_{\text{mix}} = 1$ and $b_{\text{mix}} = -\Phi < 0$.

Static OPD pipeline: divergence avoided, but absorption is poor. The static pipeline avoids gradient conflict by training each expert in isolation, removing Φ. The cost is instead paid at the consolidation stage. An OPD step trains the student on its own on-policy trajectories $y \sim \pi_\theta(\cdot \mid x)$, with teacher π_T providing a sampled-token reverse-KL policy-gradient signal [15]:

$$\begin{aligned} \mathcal{L}_{\text{OPD}}(\theta) &= -\mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{|y|} \text{sg}\big(\delta_t^{\text{OPD}}\big) \log \pi_\theta(y_t \mid s_t) \right], \\ \delta_t^{\text{OPD}} &= \log \pi_T(y_t \mid s_t) - \log \pi_\theta(y_t \mid s_t), \end{aligned} \tag{3}$$

where $s_t = (x, y_{<t})$ and sg denotes stop-gradient. This supervision is most effective when teacher and student behave similarly along on-policy trajectories. When experts have been trained to convergence in isolation, however, their behavior has drifted far from the shared base, and the OPD signal becomes hard to absorb. The resulting utility is

$$U_{\text{static}} \;\approx\; \eta(\mathcal{O}_{\text{low}}) \cdot X(D_1, D_2), \tag{4}$$

where $\eta(\mathcal{O}) \in [0, 1]$ is an absorption-efficiency function depending on teacher–student behavioral overlap $\mathcal{O}$, and $\mathcal{O}_{\text{low}}$ denotes the low overlap after experts converge in isolation. This corresponds to $a_{\text{static}} = \eta(\mathcal{O}_{\text{low}})$, positive but small, and $b_{\text{static}} = 0$.

What CoPD aims to achieve. The two paradigms sacrifice the optimization signal in opposite ways: mixed-data RLVR delivers the full signal but pays a divergence cost; static OPD removes the divergence cost but delivers only a small fraction of the signal. CoPD avoids both by (i) keeping capability-specific optimization separated to remove Φ, and (ii) maintaining teacher–student overlap where $\eta(\mathcal{O})$ is high. Its utility takes the form

$$U_{\text{CoPD}} \;\approx\; \eta(\mathcal{O}_{\text{mod}}) \cdot X(D_1, D_2), \quad \eta(\mathcal{O}_{\text{mod}}) \gg \eta(\mathcal{O}_{\text{low}}), \tag{5}$$

where $\mathcal{O}_{\text{mod}}$ is the moderate overlap actively maintained by alternating RLVR with mutual OPD. The remainder of this section grounds these claims: §3.2 introduces a measurable indicator of $\mathcal{O}$, and §3.3 shows that η does increase with $\mathcal{O}$ and that the static pipeline systematically operates at low $\mathcal{O}$.

3.2 A Behavioral Hypothesis and a Measurable Indicator

The analysis above identifies $\eta(\mathcal{O})$ as the key determinant of OPD effectiveness. To make this testable, we posit the following hypothesis.

Behavioral consistency hypothesis. An OPD signal is more easily absorbed when teacher and student exhibit similar behavioral patterns, since their on-policy trajectories then visit similar states and the teacher's predictions agree more closely with the tokens the student is likely to produce.

We instantiate this through the *top-k token overlap* along on-policy trajectories. Given a teacher π_T and a student π_θ, we define

$$\mathcal{O}_k(\pi_\theta, \pi_T) = \mathbb{E}_{x,\, y_{<t}\sim\mu_\theta} \left[\frac{|\text{Top}_k(\pi_\theta(\cdot \mid x, y_{<t})) \cap \text{Top}_k(\pi_T(\cdot \mid x, y_{<t}))|}{k} \right], \tag{6}$$

where $\text{Top}_k(\cdot)$ denotes the set of k tokens with the highest predicted probability and μ_θ is the state visitation distribution induced by the student. Larger $\mathcal{O}_k$ indicates that teacher and student agree on which tokens are likely at the states the student actually visits. The hypothesis predicts that η should rise with $\mathcal{O}_k$, at least until $\mathcal{O}_k$ becomes so high that the supervision carries no new information.

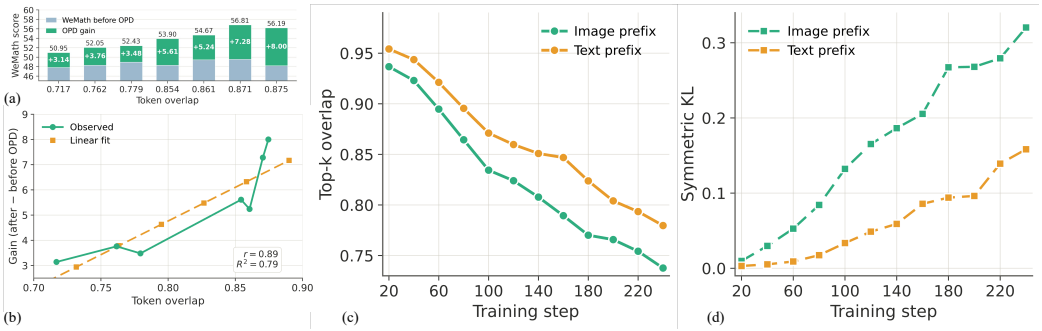

Figure 2: Pilot study motivating CoPD: post-OPD gain grows with teacher–student top-k overlap, while standard RLVR training pushes overlap in the opposite direction. **(a)** WeMath score before and after OPD across student checkpoints with different top-k overlap to the teacher; the OPD gain (green) increases with overlap. **(b)** The post-OPD gain plotted against top-k overlap, with linear fit yielding $r = 0.89$ and $R^2 = 0.79$. **(c)** As RLVR training proceeds, top-k overlap drops and symmetric KL rises, consistently across image and text branches, indicating that experts trained to convergence drift outside the effective range identified in (a) and (b).

3.3 Pilot Study

We verify the hypothesis through two complementary experiments, summarized in Figure 2. The first establishes that $\eta(\mathcal{O}_k)$ does rise with $\mathcal{O}_k$ within the regime where teacher and student differ; the second establishes that the static pipeline systematically operates at a low value of $\mathcal{O}_k$. Full setup details are in Appendix A.

Experiment 1: η rises with teacher–student overlap. We fix a teacher (an image-domain expert obtained by RLVR on D_{img}) and construct a series of students with varying $\mathcal{O}_k$ to the teacher by training the base checkpoint briefly with different sampling temperatures. For each student, we run OPD on D_{img} with identical configurations and a fixed budget, and report the WeMath score before and after distillation ($k = 10$). As shown in Figure 2(a) and (b), the post-OPD gain increases monotonically with $\mathcal{O}_k$, exhibiting a strong linear correlation ($r = 0.89$, $R^2 = 0.79$). This directly supports the behavioral consistency hypothesis: more behaviorally aligned students absorb the OPD signal more effectively. Note that this relationship must eventually reverse: when teacher and student become indistinguishable, δ_t^{OPD} in Eq. (3) approaches zero, and the OPD signal vanishes. This boundary has its own design implication: simply running OPD continuously is not enough, since teacher and student would converge in behavior and the supervisory signal would lose informativeness. An effective paradigm must therefore maintain a *moderate* behavioral gap, ensuring high enough overlap for absorption while preserving sufficient divergence to keep the supervision informative.

Experiment 2: standard RLVR drives behavioral overlap into the low-η regime. We train two independent experts via RLVR, one on D_{txt} and one on D_{img}, both from a shared base π_0, and at fixed intervals measure each expert's top-k overlap and symmetric KL with π_0 along on-policy rollouts. Treating π_0 as a proxy for the eventual student in two-stage distillation, these quantities track how teacher–student behavioral distance evolves during independent RLVR. As shown in Figure 2(c)(d), $\mathcal{O}_k$ with the shared base drops monotonically as RLVR proceeds, while symmetric KL rises by an order of magnitude; the trend is consistent across both branches. By the time experts converge, their overlap with the base is well below the levels at which η was shown to be highest in Experiment 1. This confirms that the static pipeline systematically operates in the low-η regime: experts carry rich capability-specific knowledge, but their behavioral distance to the student is too large for that knowledge to be effectively absorbed through post-hoc distillation.

Implications for method design. The two experiments together expose a structural inconsistency in the static pipeline. Experiment 1 shows that $\eta(\mathcal{O}_k)$ rises with overlap when teacher and student differ, but collapses when they become indistinguishable; Experiment 2 shows that independent RLVR drives overlap toward the low-η end. An effective paradigm must therefore satisfy three coupled requirements: (i) distillation must occur *during* expert training rather than after it, so that $\mathcal{O}_k$ does not drift toward the low-η regime; (ii) the teacher must continue to evolve with the student, so that

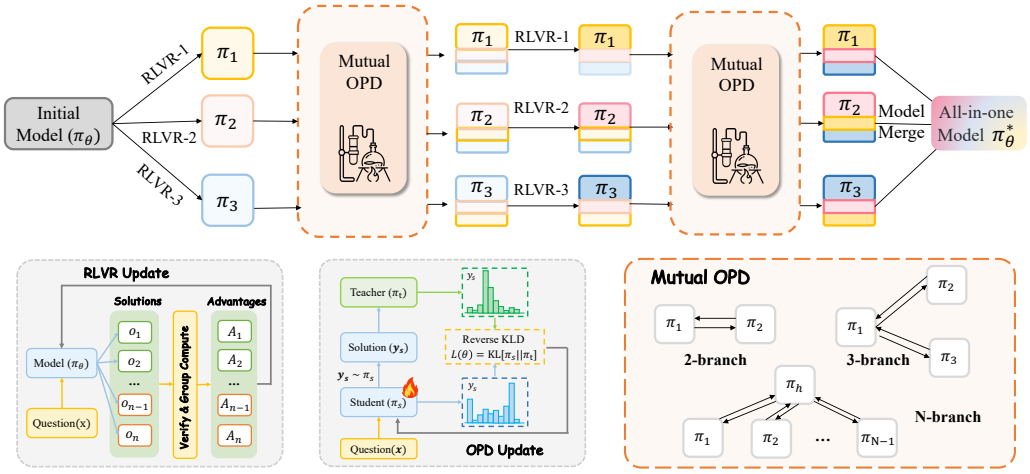

Figure 3: An overview of our CoPD method.

their overlap is actively maintained; and (iii) capability-specific training must continue to push the two sides apart, so that the supervision retains information the student does not already possess. The next section instantiates these three requirements as CoPD: cross-branch mutual OPD addresses (i) and (ii) by making each branch a continuously updated teacher for the other, while alternating with branch-specific RLVR addresses (iii) by periodically opening up the behavioral gap that subsequent OPD will then close.

4 Co-Evolving Policy Distillation

We propose **Co-Evolving Policy Distillation (CoPD)**, a framework that supports multiple expert branches to co-evolve, enabling each branch to explore its own capability through RLVR and to absorb knowledge from the others through mutual on-policy distillation at appropriate intervals. We illustrate the framework with two branches. Given a base model π_0, CoPD initializes two parallel learning branches, π_{θ_j} and π_{θ_k}, with $\theta_j^{(0)} = \theta_k^{(0)} = \theta_0$, associated with capability datasets $\mathcal{D}_j$ and $\mathcal{D}_k$, respectively. Training alternates between two phases. In the *branch-specific RLVR* phase, each branch performs reinforcement learning on its own capability data, deepening branch-specific expertise. In the *mutual OPD* phase, each branch generates on-policy rollouts and receives token-level supervision from the other branch, transferring newly acquired knowledge across capabilities. Through this alternating process, the branches maintain sufficient knowledge divergence to supply each other with informative supervision, while their behavioral distance remains close enough for that supervision to be effectively absorbed.

4.1 Branch-Specific RLVR

During the RLVR phase, each branch k independently performs GRPO on the data corresponding to capability k in order to deepen branch-specific expertise. For a prompt $x \sim \mathcal{D}_k$, branch k samples a group of rollouts $\{y_i^{(k)}\}_{i=1}^{G}$ from its current policy π_{θ_k}, which are evaluated by the capability-specific verifiable reward function r_k. The RLVR objective for branch k is defined as

$$\mathcal{L}_{\mathrm{RLVR}}^{(k)}(\theta_k) = \mathbb{E}_{x\sim\mathcal{D}_k}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(\rho_{i,t}^{(k)}\hat{A}_i^{\mathrm{RL}},\ \mathrm{clip}\left(\rho_{i,t}^{(k)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i^{\mathrm{RL}}\right)\right], \quad (7)$$

where $\hat{A}_i^{\mathrm{RL}}$ and $\rho_{i,t}^{(k)}$ follow the standard GRPO formulation. This phase gradually drives the two branches toward differentiated capability frontiers, opening up a knowledge gap between them that the subsequent mutual distillation can exploit for cross-capability transfer.

4.2 Mutual OPD

In the mutual distillation phase, each branch generates on-policy rollouts on the other branch's data and receives token-level supervision from that branch, transferring newly acquired capability-specific knowledge across branches. Because both branches start from the same base and are periodically aligned through distillation, their policies remain behaviorally close throughout training, so the supervision falls on states each branch is familiar with and can be effectively absorbed.

Concretely, branch k samples prompts from the other branch's dataset $x' \sim \mathcal{D}_j$ and generates rollouts from $\pi_{\theta_k}(\cdot \mid x')$. Branch j then evaluates these rollouts at the token level. At each position t, the teacher signal is defined as:

$$\delta_{i,t}^{(k \leftarrow j)} = \log \pi_{\theta_j}(y_{i,t}^{(k)} \mid x', y_{i,<t}^{(k)}) - \log \pi_{\theta_k}(y_{i,t}^{(k)} \mid x', y_{i,<t}^{(k)}), \tag{8}$$

and the token-level advantage for the cross-branch update is $\hat{A}_{i,t}^{(k)} = \beta_k \, \delta_{i,t}^{(k \leftarrow j)}$, where β_k balances the relative contribution of cross-branch distillation. Crucially, branch-specific RLVR does not pause during mutual OPD; instead, the two objectives are interleaved to update the model. Since the same procedure applies symmetrically, both branches alternate between the role of teacher and student at every step, continuously exchanging newly acquired knowledge.

4.3 Alternating Training Procedure

CoPD organizes the two phases described above into N alternating cycles. In each cycle $n = 1, \ldots, N$, all branches execute the following two phases in sequence, which we formulate as:

Phase I: Branch-specific RLVR. Each branch k performs S_{RL} steps of GRPO on its own capability data, as described in § 4.1:

$$\theta_k^{(n,\mathrm{I})} = \texttt{RLVR}\Big(\theta_k^{(n-1)};\, \mathcal{D}_k,\, r_k,\, S_{\mathrm{RL}}\Big)\,. \tag{9}$$

Phase II: Mutual OPD. Each branch k then performs S_{OPD} steps of mutual distillation as described in § 4.2:

$$\theta_k^{(n)} = \texttt{OPD}\Big(\theta_k^{(n,\mathrm{I})};\, \mathcal{D}_j,\, \pi_{\theta_j},\; S_{\mathrm{OPD}}\Big)\,. \tag{10}$$

The hyperparameters S_{RL} and S_{OPD} determine the rhythm between the two phases. Larger S_{RL} allows branches to accumulate more differentiated capability-specific discoveries during Phase I, providing richer signals for subsequent distillation. Larger S_{OPD} leads to more thorough knowledge transfer during Phase II. Their balance determines the overall training dynamics of CoPD, which we analyze in the ablation experiments. Because all branches start from the same base model and remain tightly coupled through continuous mutual distillation, their parameters do not diverge drastically. We exploit this by applying simple parameter merging across branches (i.e., $\theta^* = \frac{1}{K}\sum_{k=1}^{K} \theta_k$) to obtain the final unified model. Notably, even without merging, each individual branch already achieves performance comparable to domain-specific experts, as shown in our ablation study (§ 5.3). The full procedure is summarized in Algorithm 1 (Appendix F).

While we have presented CoPD with two branches for clarity, the framework scales naturally to $K > 2$ branches. To avoid full pairwise distillation, we adopt a hub-and-spoke topology, where one branch serves as a shared hub and exchanges mutual OPD with each spoke branch. In our three-branch setting (§ 5.2), the text branch acts as the hub, since the image and video branches are both vision-language models whose reasoning capabilities are grounded in the underlying LLM.

5 Experiments

5.1 Experimental Setting

Models. We evaluate CoPD on Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct [27]. Our main analysis focuses on a two-branch setting that jointly trains text and image reasoning; we additionally evaluate a three-branch setting that incorporates video reasoning to demonstrate scalability.

Table 1: Main results on image and text reasoning with Qwen3-VL-4B. Mixed RLVR runs for the combined step budget of both single-domain experts. Static OPD depends on independently trained experts and performs an additional distillation stage to consolidate them. CoPD uses the same total steps as static OPD. Best results are in **bold** and worst results (excluding Base) are marked with †.

Benchmark	Base	Image-Expert	Text-Expert	Mixed RLVR	$\text{OPD}_{V\to T}$	$\text{OPD}_{T\to V}$	**CoPD**
Image Reasoning							
MMMU	65.06	65.28	65.72	66.39	65.94	67.50	66.94
MMMU-Pro	53.03	53.89	53.05	54.28	54.09	54.71	55.10
MathVista	73.20	75.10	73.90	75.10	74.20	76.05	75.75
MathVision	55.07	55.69	55.82	56.96	55.82	56.53	57.88
$\text{ZeroBench}_{\text{sub}}$	21.41	21.41	22.01	21.86	21.41	21.71	24.40
WeMath	52.38	59.14	55.24	57.43	60.95	58.86	59.81
MathVerse	57.82	59.84	58.41	57.81	59.50	59.71	58.88
Avg.	54.00	55.76	54.88†	55.69	55.99	56.44	**56.97**
Text Reasoning							
AIME 2025	46.88	47.50	48.33	44.58	43.54	45.42	49.58
AIME 2024	58.96	55.83	60.21	57.92	59.79	58.33	60.42
HMMT 2025	25.83	21.67	27.50	24.17	25.83	26.67	30.83
MATH-500	92.80	93.90	93.65	93.55	93.45	93.40	94.50
Minerva Math	54.41	58.64	59.74	57.17	58.55	56.62	58.46
Avg.	55.78	55.51	57.89	55.48†	56.23	56.09	**58.76**
Overall Avg.	54.74	55.65	56.13	55.60†	56.09	56.29	**57.71**

Data. For text reasoning, we use Polaris-Dataset-53K [28]; for image reasoning, MMFineReason-123K [29]; for video reasoning, 40K samples filtered from OneThinker [30], VideoChat-R1 [31], and Video-R1 [9]. Full details on data construction and filtering are provided in Appendix B.

Benchmarks. Image reasoning is evaluated on seven benchmarks (MMMU [32], MMMU-Pro [33], MathVista [34], MathVision [35], $\text{ZeroBench}_{\text{sub}}$ [36], WeMath [37], MathVerse [38]), text reasoning on five (AIME 2024/2025 [39], HMMT 2025 February [40], MATH-500 [41], Minerva Math [42]), and video reasoning on four (Video-Holmes [43], MVBench [44], MMVU [45], VideoMathQA [46]). We report accuracy (%) throughout, with domain averages denoted Vis. Avg., Text Avg., and Video Avg.

Baselines. We compare CoPD against: **Domain-specific Experts** are trained independently from the base model via RLVR. **Mixed RLVR** combines all data into a single pool and trains one model. **Static OPD** (V→T / T→V) first trains domain experts via RLVR independently, then one expert serves as a fixed teacher and transfers knowledge to the other through OPD. We report results for both directions, where V→T denotes the image expert teaching the text branch and T→V denotes the reverse. For the three-branch setting, we use **multi-teacher OPD (MOPD)** as a stronger baseline, where all domain experts jointly distill into a single student.

5.2 Main Results

Co-Evolution on Text and Image Reasoning. Table 1 presents the main results on text and image reasoning, where CoPD achieves the best overall performance. Mixed RLVR weakens text reasoning compared to the Text-Expert, confirming the capability divergence cost analyzed in §3. Static OPD mitigates this interference by training experts separately, but the teacher's capability is poorly absorbed. For example, in the T→V direction, the Text-Expert's strong text capability (57.89) drops to 56.09 in the student. This is consistent with the analysis in § 3: when experts have drifted far from the student, their thinking patterns diverge substantially, making the distillation signal difficult to absorb. In contrast, CoPD improves both image and text reasoning simultaneously, surpassing the experts on both sides. Results on Qwen3-VL-8B confirm the same trend (Appendix C.3).

Scaling to Co-Evolution on Text, Image, and Video Reasoning. Table 2 extends CoPD to a three-branch setting that jointly consolidates text, image, and video reasoning. CoPD again achieves the best overall performance, outperforming MOPD across all capability groups. Notably, MOPD underperforms the video and text experts, confirming that static multi-teacher distillation struggles

Table 2: Average results on image, text, and video reasoning benchmarks with Qwen3-VL-4B. See Appendix C.2 for benchmark-level results. Overall Avg. averages all benchmarks.

Benchmark	Base	Image-Expert	Text-Expert	Video-Expert	Mixed RLVR	MOPD	**CoPD**
Image Avg.	54.00	55.76	54.88	54.71†	56.17	56.37	**57.12**
Text Avg.	55.78	55.51	57.89	56.84	55.39†	56.80	**58.63**
Video Avg.	56.22	58.27	55.54†	58.75	**59.62**	58.32	59.21
Overall Avg.	55.11	56.31	55.98†	56.39	56.79	56.99	**58.12**

Table 3: Ablation results on image and text reasoning benchmarks. I-OPD/T-OPD denotes distillation from the image/text branch (teacher); Text-Branch Only and Image-Branch Only denote removing the merge operation and using only the text or image branch for evaluation, respectively.

Method	*Image Reasoning*								*Text Reasoning*						All
	MMMU	M-P.	Vista	Vision	ZB_s	WeM.	Verse	Avg.	A'25	A'24	H'25	M500	Min.	Avg.	
CoPD	66.94	55.10	75.75	57.88	24.40	59.81	58.88	56.97	49.58	60.42	30.83	94.50	58.46	58.76	57.71
w/o I-OPD	67.39	55.49	75.30	57.35	21.71	61.71	58.50	56.78	47.50	59.79	29.17	93.25	57.35	57.41	57.04
w/o T-OPD	66.00	54.78	75.00	57.66	22.01	61.33	58.59	56.48	49.38	57.71	28.33	94.10	59.38	57.78	57.02
Text-Branch Only	65.83	54.25	76.10	56.86	24.85	57.62	58.28	56.26	48.96	62.08	30.83	93.85	57.35	58.61	57.24
Image-Branch Only	65.39	54.82	76.15	56.99	24.40	60.29	59.39	56.78	49.38	59.58	25.00	94.10	57.81	57.17	56.94

as the number of branches grows, while Mixed RLVR improves video reasoning at the cost of text reasoning, exhibiting the same divergence pattern as the two-branch setting. CoPD avoids both failure modes by co-evolving all three branches. See Appendix C.2 for full results and detailed analysis.

5.3 Analysis

Ablation Study. Table 3 studies the contribution of each component in CoPD. Removing I-OPD or T-OPD causes the corresponding cross-domain reasoning to drop (text: 58.76→57.41 without I-OPD; image: 56.97→56.48 without T-OPD), confirming that mutual OPD in both directions is necessary. We also ablate the merge operation: retaining only one branch preserves that branch's strength but weakens the other domain. Notably, even without merging, each single branch already surpasses both static OPD variants in Table 1 on overall performance (57.24 and 56.94 vs. 56.09 and 56.29), showing that co-evolution alone produces well-rounded branches. Merging further consolidates their complementary strengths, yielding the best overall result.

Behavioral Pattern Consistency During Training. Figures 4 (a) and (b) track the top-k token overlap and symmetric KL between the two branches throughout training. In the baseline, both metrics diverge monotonically, while symmetric KL rises by an order of magnitude. Since static OPD applies distillation after this expert training completes, it operates at the point where the two experts are furthest apart, confirming the low absorption efficiency predicted in §3. In CoPD, top-k overlap decreases during each RLVR phase but recovers during mutual OPD, staying above 0.90 throughout training, and KL remains consistently low. This confirms the core design: RLVR creates divergence that makes distillation informative, and mutual OPD restores alignment that makes it absorbable.

Effect of the $S_{\mathrm{RL}}/S_{\mathrm{OPD}}$ Ratio. We further analyze the impact of the ratio between RLVR exploration and OPD consolidation steps in Figure 4 (c). CoPD consistently outperforms static OPD under different $S_{\mathrm{RL}}/S_{\mathrm{OPD}}$ ratios. Among different ratios, S_{RL}:S_{OPD} = 1.5:1 achieves the best overall performance, suggesting that sufficient branch-specific exploration is needed to create useful complementary knowledge, while overly long exploration may weaken the alignment between branches and reduce distillation effectiveness. We therefore adopt this ratio throughout all experiments.

6 Conclusion

This paper explores a core question: how can we better absorb the capabilities of multiple experts into a single model? We identify that both conventional approaches, mixed RLVR and the traditional static OPD pipeline, suffer from non-negligible loss of expert capabilities. We propose a central idea: Co-evolving Policy Distillation, where distillation among experts should occur *during* training rather than *after* it, and multiple expert models should serve as teachers and students to one another, co-

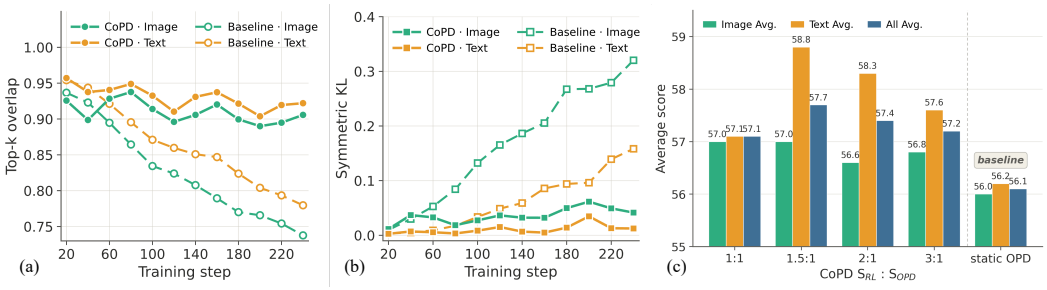

Figure 4: Training dynamics and design analysis of CoPD. **(a)** Top-k token overlap between the two branches throughout training. **(b)** Symmetric KL between the two branches. **(c)** Effect of the S_{RL}:S_{OPD} ratio on final performance, with 1.5:1 achieving the best overall accuracy.

evolving in synergy. Experiments validate that CoPD achieves an all-in-one consolidation of multiple expert capabilities, even outperforming the respective expert models. The strong performance of CoPD suggests an intriguing perspective: *model parallel training* may serve as a promising scaling paradigm for further broadening the boundaries of model capabilities.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62472419, 62472420).

References

[1] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[2] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

[3] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, 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, 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, 2025. URL https://arxiv.org/abs/2503.14476.

[4] 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, 2025. URL https://arxiv.org/abs/2507.18071.

[5] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models, 2026. URL https://arxiv.org/abs/2503.06749.

[6] Huanjin Yao, Qixiang Yin, Jingyi Zhang, Min Yang, Yibo Wang, Wenhao Wu, Fei Su, Li Shen, Minghui Qiu, Dacheng Tao, and Jiaxing Huang. R1-sharevl: Incentivizing reasoning capability of multimodal large language models via share-grpo, 2025. URL https://arxiv.org/abs/2505.16673.

[7] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, 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, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, 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, 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, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, 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, Yanhong Xu, 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: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

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

[9] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[10] Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, Xiangnan Fang, Zewen He, Zhenbo Luo, Wenxuan Wang, Junqi Lin, Jian Luan, and Qin Jin. Time-r1: Post-training large vision language model for temporal video grounding, 2025. URL `https://arxiv.org/abs/2503.13377`.

[11] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Angang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Yuanying Guo, Xiaoru Hao, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li, Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Yue Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Yingwei Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Ao Wang, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang Wang, Feng Wang, Hailong Wang, Haiming Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie

Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Yifei Xin, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Ruihan Yang, Xiaofei Yang, Xinlong Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Wenjie Ye, Zhuorui Ye, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi k2.5: Visual agentic intelligence, 2026. URL https://arxiv.org/abs/2602.02276.

[12] GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

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

[14] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18878–18890. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9d27fdf2477ffbff837d73ef7ae23db9-Paper.pdf.

[15] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

[16] DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL `https://arxiv.org/abs/2606.19348`.

[17] OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu,

Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

[18] Linhao Yu, Tianmeng Yang, Siyu Ding, Renren Jin, Naibin Gu, Xiangzhao Hao, Shuaiyi Nie, Deyi Xiong, Weichong Yin, Yu Sun, and Hua Wu. Knowrl: Boosting llm reasoning via reinforcement learning with minimal-sufficient knowledge guidance, 2026. URL https://arxiv.org/abs/2604.12627.

[19] Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models, 2025. URL https://arxiv.org/abs/2505.07686.

[20] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

[21] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

[22] Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation, 2026. URL https://arxiv.org/abs/2602.12125.

[23] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026. URL https://arxiv.org/abs/2601.18734.

[24] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

[25] Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level, 2026. URL https://arxiv.org/abs/2605.06387.

[26] Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026. URL https://arxiv.org/abs/2604.13016.

[27] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

[28] Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

[29] Honglin Lin, Zheng Liu, Yun Zhu, Chonghan Qin, Juekai Lin, Xiaoran Shang, Conghui He, Wentao Zhang, and Lijun Wu. Mmfinereason: Closing the multimodal reasoning gap via open data-centric methods. *arXiv preprint arXiv:2601.21821*, 2026.

[30] Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, et al. Onethinker: All-in-one reasoning model for image and video. *arXiv preprint arXiv:2512.03043*, 2025.

[31] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2504.06958.

[32] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL https://arxiv.org/abs/2311.16502.

[33] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2025. URL https://arxiv.org/abs/2409.02813.

[34] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024. URL https://arxiv.org/abs/2310.02255.

[35] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=QWTCcxMpPA.

[36] Jonathan Roberts, Mohammad Reza Taesiri, Ansh Sharma, Akash Gupta, Samuel Roberts, Ioana Croitoru, Simion-Vlad Bogolin, Jialu Tang, Florian Langer, Vyas Raina, Vatsal Raina, Hanyi Xiong, Vishaal Udandarao, Jingyi Lu, Shiyang Chen, Sam Purkis, Tianshuo Yan, Wenye Lin, Gyungin Shin, Qiaochu Yang, Anh Totti Nguyen, David I. Atkinson, Aaditya Baranwal, Alexandru Coca, Mikah Dang, Sebastian Dziadzio, Jakob D. Kunz, Kaiqu Liang, Alexander Lo, Brian Pulfer, Steven Walton, Charig Yang, Kai Han, and Samuel Albanie. Zerobench: An impossible visual benchmark for contemporary large multimodal models, 2025. URL https://arxiv.org/abs/2502.09696.

[37] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Yifan Zhang, Xiao Zong, Yida Xu, Muxi Diao, Zhimin Bao, Chen Li, and Honggang Zhang. We-math: Does your large multimodal model achieve human-like mathematical reasoning?, 2024. URL https://arxiv.org/abs/2407.01284.

[38] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?, 2024. URL https://arxiv.org/abs/2403.14624.

[39] MAA Committees. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

[40] Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, February 2025. URL https://matharena.ai/.

[41] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

[42] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

[43] Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. Video-holmes: Can mllm think like holmes for complex video reasoning?, 2025. URL https://arxiv.org/abs/2505.21374.

[44] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2024. URL https://arxiv.org/abs/2311.17005.

[45] Yilun Zhao, Lujing Xie, Haowei Zhang, Guo Gan, Yitao Long, Zhiyuan Hu, Tongyan Hu, Weiyuan Chen, Chuhan Li, Junyang Song, Zhijian Xu, Chengye Wang, Weifeng Pan, Ziyao Shangguan, Xiangru Tang, Zhenwen Liang, Yixin Liu, Chen Zhao, and Arman Cohan. Mmvu: Measuring expert-level multi-discipline video understanding, 2025. URL https://arxiv.org/abs/2501.12380.

[46] Hanoona Rasheed, Abdelrahman Shaker, Anqi Tang, Muhammad Maaz, Ming-Hsuan Yang, Salman Khan, and Fahad Shahbaz Khan. Videomathqa: Benchmarking mathematical reasoning via multimodal understanding in videos, 2025. URL https://arxiv.org/abs/2506.05349.

[47] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

[48] Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. Areal: A large-scale asynchronous reinforcement learning system for language reasoning, 2025. URL https://arxiv.org/abs/2505.24298.

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

[50] Chuanyu Qin, Chenxu Yang, Qingyi Si, Naibin Gu, Dingyu Yao, Zheng Lin, Peng Fu, Nan Duan, and Jiaqi Wang. Easyvideor1: Easier rl for video understanding, 2026. URL https://arxiv.org/abs/2604.16893.

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

[52] Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, Yuwen Xiong, and Richong Zhang. Easyr1: An efficient, scalable, multi-modality rl training framework. https://github.com/hiyouga/EasyR1, 2025.

[53] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models, 2023. URL https://arxiv.org/abs/2306.01708.

[54] Fahim Tajwar, Yiding Jiang, Abitha Thankaraj, Sumaita Sadia Rahman, J Zico Kolter, Jeff Schneider, and Ruslan Salakhutdinov. Training a generally curious agent, 2025. URL https://arxiv.org/abs/2502.17543.

[55] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models, 2025. URL https://arxiv.org/abs/2505.24864.

[56] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL https://arxiv.org/abs/2502.01456.

[57] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

A Pilot Study Details

Experiment 1: OPD gain vs. teacher–student overlap. The teacher is an image-domain expert obtained by running GRPO on D_{img} for 240 steps from Qwen3-VL-4B-Instruct. To construct students with varying $\mathcal{O}_k$, we take the same base checkpoint and run 20 steps of GRPO on D_{img} under sampling temperatures $\tau \in \{0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2\}$. Different temperatures induce different exploration patterns, producing checkpoints whose top-k distributions differ from the teacher to varying degrees. For each student, $\mathcal{O}_k$ (k=10) is measured over 512 on-policy rollouts from D_{img}. OPD is then run for 40 steps with learning rate 1×10^{-6}, rollout batch size 256, 8 rollouts per prompt, and sampling temperature 1.0 for all students. The varying temperatures are used only in the student construction phase to induce different levels of behavioral overlap and are not carried into the distillation stage. WeMath accuracy is evaluated before and after OPD.

Experiment 2: behavioral drift during independent RLVR. Two experts are trained independently from the shared base π_0 (Qwen3-VL-4B-Instruct): one on D_{txt} and one on D_{img}, using learning rate 1×10^{-6}, rollout batch size 256, 8 rollouts per prompt, and temperature 1.0. At every 20 training steps, we save a checkpoint and compute two metrics over 512 on-policy rollouts: (1) top-k overlap $\mathcal{O}_k$ (k=10) between the current expert and π_0, and (2) symmetric KL divergence $\frac{1}{2}[D_{\text{KL}}(\pi_{\text{expert}} \| \pi_0) + D_{\text{KL}}(\pi_0 \| \pi_{\text{expert}})]$.

B Experimental Setting Details

Training Data and Evaluation Benchmarks We evaluate CoPD on its ability to co-evolve text, image, and video reasoning capabilities through parallel branch training. Our main analysis focuses on the two-branch setting with text and image reasoning; we additionally evaluate a three-branch setting that incorporates video reasoning to demonstrate scalability. For text reasoning, we use Polaris-Dataset-53K [28], filtered from DeepScaleR-Preview-Dataset [47] and AReal-boba-Data [48] to retain high-quality mathematical reasoning problems. For image reasoning, we use MMFineReason-123K [29], a collection of image reasoning samples with verifiable answers. For video reasoning, we collect training data from OneThinker [30], VideoChat-R1 [31], and Video-R1 [9], and filter with Qwen3-8B-VL by removing samples with a pass rate of either 0% or 100%, retaining 40K samples of moderate difficulty.

Evaluation Benchmarks For evaluation, image reasoning is assessed on seven benchmarks. MMMU [32] and MMMU-Pro [33] cover multi-discipline college-level problems. MathVista [34] and MathVision [35] test mathematical reasoning with visual context. ZeroBench [36] evaluates zero-shot visual reasoning. WeMath [37] and MathVerse [38] focus on math problems presented with diagrams and multi-format visual inputs. Text reasoning is assessed on five benchmarks. AIME 2024 and AIME 2025 [39] are competition-level mathematical olympiad problems. HMMT 2025 [40] contains problems from the Harvard-MIT Mathematics Tournament. MATH-500 [41] is a curated subset of the MATH benchmark, and Minerva Math [42] covers scientific and quantitative reasoning. For all benchmarks, we report accuracy (%) as the evaluation metric. We report Vis. Avg. and Text Avg. as the mean accuracy over the respective benchmark groups. For the three-branch setting, we additionally evaluate video reasoning on Video-Holmes [43], MVBench [44], MMVU [45], and VideoMathQA [46], and report Video Avg. as the mean accuracy over these video benchmarks. All evaluations are conducted using EvalScope [49] for text and image reasoning benchmarks and EasyVideoR1 [50] for video reasoning benchmarks. For all text and image benchmark evaluations, the maximum generated response length is set to 16,384 tokens. To reduce sampling variance during evaluation, we report accuracy averaged over repeated independent responses. We sample 16 responses per problem for smaller benchmarks and 4 for larger benchmarks, using the same protocol for all methods and model sizes.

Models and Baselines We mainly conduct experiments on Qwen3-VL-4B-Instruct [27] and compare CoPD against the following baselines. Text-Expert and Image-Expert are trained independently from the base model via RLVR on text reasoning data and image reasoning data, respectively. In the three-branch setting, we additionally train a Video-Expert on video reasoning data. Mixed RLVR combines all data into a single pool and trains one model via RLVR. Static OPD (V→T / T→V) follows a two-stage pipeline: two branches are first independently trained via RLVR on their respective domains, then one expert serves as a fixed teacher and transfers knowledge to the other

Table 4: Compute cost under matched branch-update budgets in the two-branch Qwen3-VL-8B setting.

Method	Training composition	Updates	Wall-clock	GPU-hours	Peak memory/GPU
Mixed RLVR	440 mixed-domain updates	440	66 h	2,112	≈133 GB
Static OPD	$200 + 200 + 40$	440	47 h	1,504	≈133 GB
CoPD	220×2	440	52 h	1,664	≈133 GB

through unidirectional OPD. We report results for both directions, where V→T denotes the image expert teaching the text branch and T→V denotes the reverse. For the three-branch setting, we use multi-teacher OPD (MOPD) as a stronger baseline, where all domain experts jointly distill into a single student. CoPD (Ours) alternates between RLVR and bidirectional mutual on-policy distillation throughout training.

Implementation Details We implement CoPD on top of the EasyVideoR1 framework [50], which builds upon verl [51] and EasyR1 [52]. During training, the maximum input and output length are both set to 16,384 tokens. The learning rate is fixed at 1×10^{-6}. The rollout batch size is set to 256, and for each prompt we sample 8 rollouts at temperature 1.0. The clipping bounds are set to $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. The cross-branch distillation weight β_k is set to 0.01 for all branches. Static OPD is built on top of two independently trained specific experts and performs one additional stage of OPD. For clarity, we count one update to one model branch as one branch-update step. Each domain-specific expert is trained for 200 branch-update steps. For each static OPD direction, we initialize from the two independently trained experts and perform 40 additional OPD steps to ensure convergence, resulting in 440 branch-update steps in total. CoPD uses the same total budget: in the two-branch setting, each branch is trained for 220 branch-update steps, yielding 440 branch-update steps in total. Mixed RLVR is trained for 440 update steps with balanced sampling across the two domains, ensuring matched data throughput across methods. All experiments are conducted on 32 NVIDIA H200 GPUs.

C Extended Experiments

C.1 Computational Cost

We report compute costs for the two-branch Qwen3-VL-8B setting. All methods use the same allocation of 32 NVIDIA H200 GPUs and the same total budget of 440 branch-update steps. GPU-hours are calculated as wall-clock time multiplied by 32. As shown in Table 4, the peak memory usage is similar across methods because it is dominated by the vLLM rollout engine and the shared rollout configuration. CoPD therefore uses the same update budget as the baselines, although its wall-clock time is 52 hours, compared with 47 hours for Static OPD and 66 hours for Mixed RLVR.

Regarding branch residency, CoPD does not require all branches to be loaded on the same devices. In our implementation, branches are assigned to disjoint GPU groups and processed concurrently. With fewer GPUs, the two directions of mutual OPD can instead be executed sequentially by caching the student rollouts and teacher token log-probabilities before applying the corresponding updates. This reduces the peak GPU requirement at the cost of additional I/O and wall-clock time.

C.2 Scaling to Three Branches

Table 5 extends CoPD from the dual-branch setting to a three-branch setting, where text, image, and video reasoning capabilities are jointly optimized and consolidated. The results follow the same trend as Table 1: CoPD achieves the best overall performance and improves over MOPD across major capability groups, showing that its effectiveness generalizes beyond pairwise capability transfer. Notably, MOPD underperforms the domain-specific experts on both video reasoning (58.32 vs. 58.75) and text reasoning (56.80 vs. 57.89), confirming that static multi-teacher distillation struggles to absorb all experts' knowledge as the number of capability branches grows. Mixed RLVR achieves the highest video average among baselines, likely because video reasoning benefits from diverse data, but this comes at the cost of text reasoning (55.39), again exhibiting the capability divergence pattern

Table 5: Main results on image, text, and video reasoning benchmarks with Qwen3-VL-4B. Mixed RLVR runs for the combined step budget of all single-domain experts. MOPD depends on independently trained experts and performs an additional distillation stage to consolidate them. CoPD uses the same total steps as MOPD. Best results are in **bold** and worst results (excluding the Base Model) are marked with †.

Benchmark	Base	Image-Expert	Text-Expert	Video-Expert	Mixed RLVR	MOPD	**CoPD**
Image Reasoning							
MMMU	65.06	65.28	65.72	66.17	66.11	66.61	67.78
MMMU-Pro	53.03	53.89	53.05	53.08	54.06	54.64	55.12
MathVista	73.20	75.10	73.90	74.85	75.55	74.75	76.80
MathVision	55.07	55.69	55.82	55.30	56.00	56.99	57.57
$\text{ZeroBench}_{\text{sub}}$	21.41	21.41	22.01	22.01	22.60	22.75	23.35
WeMath	52.38	59.14	55.24	53.43	59.05	59.90	59.90
MathVerse	57.82	59.84	58.41	58.16	59.82	58.93	59.34
Avg.	54.00	55.76	54.88	54.71†	56.17	56.37	**57.12**
Text Reasoning							
AIME 2025	46.88	47.50	48.33	47.29	45.62	46.04	49.38
AIME 2024	58.96	55.83	60.21	56.46	54.17	60.83	60.42
HMMT 2025	25.83	21.67	27.50	30.83	25.83	25.00	31.67
MATH-500	92.80	93.90	93.65	93.45	92.95	93.85	94.35
Minerva Math	54.41	58.64	59.74	56.16	58.36	58.27	57.35
Avg.	55.78	55.51	57.89	56.84	55.39†	56.80	**58.63**
Video Reasoning							
Video-Holmes	43.28	44.47	42.19	45.35	45.62	43.33	43.77
MVBench	68.27	67.82	65.73	70.86	67.88	67.71	69.16
MMVU(mc)	64.00	67.68	67.04	67.84	69.28	68.80	68.16
VideoMathQA	49.33	53.10	47.19	50.95	55.71	53.43	55.76
Avg.	56.22	58.27	55.54†	58.75	**59.62**	58.32	59.21
Overall Avg.	55.11	56.31	55.98†	56.39	56.79	56.99	**58.12**

observed in the two-branch setting. CoPD avoids this trade-off by co-evolving all three branches, consolidating their capabilities without sacrificing any individual domain.

C.3 Scaling to Larger Model Scale

Table 6 reports results on Qwen3-VL-8B-Instruct, where CoPD again achieves the best overall performance, outperforming all baselines by a clear margin. Mixed RLVR exhibits the same seesaw effect observed at the 4B scale: it achieves a competitive image average but the worst text average among all methods (58.53), falling well below the Text-Expert (60.43), confirming that capability divergence is not resolved by scaling up model capacity. Static OPD mitigates this interference by training experts separately, but the teacher's capability remains poorly absorbed. In the T→V direction, the Text-Expert's text capability (60.43) drops to 59.01 in the student, consistent with the absorption inefficiency identified in §3. In contrast, CoPD improves both image and text reasoning simultaneously, surpassing the domain-specific experts on both sides (image: 60.89 vs. 59.35; text: 61.53 vs. 60.43). These results confirm that the findings at the 4B scale generalize to larger models: co-evolution consistently outperforms both mixed RLVR and static OPD pipelines.

C.4 Merging Controls

CoPD obtains the final unified model by equal-weight averaging the parameters of the co-evolved branches. We first examine this choice through linear interpolation and then compare CoPD with post-hoc merging baselines.

Table 6: Main results on image and text reasoning benchmarks with Qwen3-VL-8B. All settings follow the same protocol as Table 1. Best results are in **bold** and worst results (excluding the Base Model) are marked with †.

Benchmark	Base	Image-Expert	Text-Expert	Mixed RLVR	OPD$_{V\to T}$	OPD$_{T\to V}$	**CoPD**
Image Reasoning							
MMMU	68.61	70.22	70.11	69.33	69.33	69.72	70.33
MMMU-Pro	57.30	59.40	57.47	59.12	59.91	59.03	60.25
MathVista	78.40	78.75	77.15	78.70	78.35	79.05	79.30
MathVision	57.75	59.90	58.83	59.84	61.05	59.54	61.60
ZeroBench$_{\text{sub}}$	23.50	22.75	25.90	24.10	23.65	23.80	26.20
WeMath	56.76	64.48	58.48	68.95	61.90	67.71	67.05
MathVerse	57.37	59.98	58.57	60.93	60.93	60.43	61.49
Avg.	57.10	59.35	58.07†	60.14	59.30	59.90	**60.89**
Text Reasoning							
AIME 2025	47.50	48.75	51.46	46.88	49.79	48.13	50.83
AIME 2024	59.58	62.71	62.71	61.04	63.75	60.83	66.25
HMMT 2025	33.33	30.83	35.00	30.83	35.00	31.67	35.83
MATH-500	94.10	94.45	94.15	94.50	94.15	94.20	94.55
Minerva Math	55.15	59.19	58.82	59.38	59.01	60.20	60.20
Avg.	57.93	59.19	60.43	58.53†	60.34	59.01	**61.53**
Overall Avg.	57.45	59.28	59.05†	59.47	59.74	59.53	**61.16**

Table 7: Linear interpolation between the final text and image branches of CoPD with Qwen3-VL-4B. Ratios denote the relative weights of the text and image branches, respectively.

Text:Image	9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9
Image Avg.	55.72	55.36	55.87	56.04	56.97	56.82	57.06	56.34	56.83
Text Avg.	58.63	57.99	57.66	58.56	58.76	57.35	57.92	57.23	57.25

Table 8: Direct merging controls in the two-branch setting with Qwen3-VL-4B. Best results are in **bold**.

Method	Image Avg.	Text Avg.	Overall
Direct merging (Averaging)	55.37	55.94	55.61
Direct merging (TIES)	56.15	56.42	56.26
CoPD	**56.97**	**58.76**	**57.71**

Linear interpolation. We linearly interpolate the final text and image branches as

$$\theta(\lambda) = \lambda\theta_{\text{text}} + (1-\lambda)\theta_{\text{image}}. \tag{11}$$

Here, λ controls the weight of the text branch, and we evaluate $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$. The results for different text-to-image ratios are shown in Table 7. The 5:5 midpoint provides a good balance between the two domains. Given the cost of evaluating different merging weights in this post-training setting, equal-weight averaging provides a strong parameter-free choice without requiring additional hyperparameter tuning.

Post-hoc merging baselines. We compare CoPD with direct averaging and TIES-Merging [53]. Direct averaging uses independently trained branches with the same initialization and task-specific RLVR setup, but without intermediate mutual OPD. TIES-Merging is applied to the same independently trained branches as a stronger post-hoc merging baseline. The results are reported in Table 8. CoPD outperforms both methods on all three metrics, highlighting the contribution of bidirectional mutual OPD to transferring complementary capabilities before the final merge.

Table 9: Results on mathematical reasoning and code generation with Qwen3-VL-4B. Overall averages the four benchmark scores. Best results are in **bold**.

Method	Minerva	AIME25	Math Avg.	HumanEval+	MBPP+	Code Avg.	Overall
Base	54.41	46.88	50.65	79.88	67.00	73.44	62.04
Math Expert	**59.74**	48.33	**54.04**	80.03	67.39	73.71	63.87
Code Expert	56.16	45.21	50.69	82.62	68.19	75.41	63.05
Mixed RLVR	57.08	47.92	52.50	83.08	68.32	75.70	64.10
Static OPD (Math→Code)	58.82	47.08	52.95	81.86	67.86	74.86	63.91
CoPD	59.28	**48.54**	53.91	**84.15**	**68.78**	**76.47**	**65.19**

C.5 Co-Evolution on Math and Code

Our main experiments consolidate capabilities across text, image, and video domains. Prior work suggests that training on relatively similar tasks can promote transfer and generalization [54, 55], raising the question of whether CoPD remains beneficial when the domains are more closely related and joint training can exploit shared reasoning structures. We therefore examine mathematical reasoning and code generation, which share a textual modality and algorithmic structures such as decomposition and planning.

Both branches start from Qwen3-VL-4B-Instruct. The Math Expert is the same Text Expert used in our main experiments, which was trained via RLVR on Polaris-Dataset-53K. The Code Expert is trained on Eurus-RL-Code [56] with the same optimization configuration. We evaluate on Minerva Math, AIME 2025, HumanEval+, and MBPP+ [57]. All unified methods use matched total branch-update budgets, and Static OPD uses the Math→Code direction.

Table 9 shows that CoPD achieves the best overall performance, closely matching the Math Expert on mathematics and surpassing the Code Expert on code generation. Mixed RLVR also improves code generation over the Code Expert, and its overall gap to CoPD is smaller than in the text–image setting. However, it remains below the Math Expert on mathematics, while Static OPD falls below the corresponding experts in both domains. These observations are compatible with the benefits of multi-task training reported in prior work: positive transfer can coexist with capability trade-offs. CoPD improves the balance between the two capabilities through intermediate mutual OPD.

D Limitations

While we have validated CoPD in the two-branch setting and demonstrated that it scales to three branches covering text, image, and video reasoning, its effectiveness with a larger number of branches remains to be explored. That said, the strong performance of CoPD suggests an intriguing perspective: *model parallel training* may serve as a promising scaling paradigm for further broadening the boundaries of model capabilities.

E Broader Impact

CoPD offers an efficient approach to consolidating multiple capabilities into a single model, which helps reduce deployment costs and lower energy consumption and carbon emissions during inference. It also makes versatile models more accessible to resource-limited teams, promoting the democratization of AI technology. However, more capable all-in-one models could be misused to generate misleading content. Furthermore, as a general post-training method, CoPD could amplify safety risks if applied to consolidate harmful capabilities (e.g., simultaneously enhancing malicious code generation and social engineering). That said, these risks are not unique to CoPD but are common challenges shared by all capability-enhancement methods.

F Algorithm of CoPD

Here, we present the complete algorithm of the proposed method in Algorithm 1.

Algorithm 1 CoPD: Co-Evolving Policy Distillation

Require: Base model π_{θ_0}, K capability datasets $\{\mathcal{D}_k\}_{k=1}^{K}$, reward functions $\{r_k\}_{k=1}^{K}$, total cycles N, RLVR steps S_{RL}, OPD steps S_{OPD}, OPD neighbor sets $\{\mathcal{N}(k)\}_{k=1}^{K}$
// For $K = 2$ branches, $\mathcal{N}(k)$ contains the other branch.
// For $K > 2$ branches, $\mathcal{N}(k)$ follows a hub-and-spoke topology.
Initialize K branches: $\theta_k \leftarrow \theta_0$ for all k
for $n = 1, \ldots, N$ **do**
  // Phase I: Branch-specific RLVR (all branches in parallel)
  for branch $k = 1, \ldots, K$ **in parallel do**
    Optimize θ_k on $\mathcal{D}_k$ with GRPO (Eq. 7) for S_{RL} steps
  end for
  // Phase II: Mutual OPD (all branches in parallel)
  for branch $k = 1, \ldots, K$ **in parallel do**
    for $s = 1, \ldots, S_{\mathrm{OPD}}$ **do**
      Native: generate rollouts on $\mathcal{D}_k$, update with GRPO using r_k
      Cross-branch: for each $j \in \mathcal{N}(k)$, generate rollouts on $\mathcal{D}_j$ from π_{θ_k}; compute teacher signal $\delta^{(k \leftarrow j)}$ from π_{θ_j} (Eq. 8); set $\hat{A}^{(k)} = \beta_k \, \delta^{(k \leftarrow j)}$
      Combine native and cross-branch batches; update θ_k
    end for
  end for
end for
// Merge co-evolved branches into a unified model
$\theta^* \leftarrow \texttt{Merge}(\theta_1, \theta_2, \ldots, \theta_K)$
return θ^*

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction (Section 1) accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses the limitations of the approach in the Appendix D.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper is mainly based on observation, making conjectures and methods and proving the effects through experiments.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides a detailed description of the experimental data setup and hyperparameter settings in Section 5 and the Appendix B.

   Guidelines:

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

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

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

Answer: [No]

Justification: The datasets, baseline methods, and models used in the paper are fully open-source and available on Hugging Face. The paper includes the key implementation steps and code in Section 4 and the Appendix F.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The paper provides a detailed description of the experimental data setup and hyperparameter settings in Section 5 and the Appendix B.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: RLVR training is highly resource-intensive. In our setting, we train multiple branches across three different modalities, which can take several days on a multi-GPU setup. Besides, most recent RLVR studies, such as Dr. GRPO, DeepScaleR, and R1-ShareVL etc., also runs the training only once. Nevertheless, we evaluate CoPD across multiple model sizes, capability settings, benchmarks, and ablation variants. The consistent improvements in Section 5 and Appendix C provide cross-validation for the effectiveness of our method.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report them in Appendix A and Appendix C.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We report them in the Appendix E.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We report them in the Appendix E.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper presents an improved approach based on the existing model architecture. The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: This article uses assets reasonably in compliance with the license, and the assets used are cited in the article.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [No]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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