GIFT: Games as Informal Training for Generalizable LLMs

Anonymous ACL submission

Abstract

Recent LLMs excel at formal tasks such as mathematical reasoning and code generation, but still struggle with broader abilities such as planning, creativity, and social intelligence. Inspired by human learning, where formal instruction and informal experience jointly shape intelligence, we introduce informal learning into LLM training and use games as annotation-free, feedback-driven environments. To cover diverse abilities including abstract reasoning, planning, creativity, and social interaction, we combine formal math tasks with three representative game tasks, including Matrix Games, TicTacToe, and Who's the Spy. However, directly mixing these tasks under a unified RL objective can blur task-specific learning signals and provides no explicit guidance for coordinating task-gradient directions. To combat these, we propose Coordinated Subtask Training (CST), which replaces a single mixed update with sequential subtask-specific updates, separating heterogeneous RL signals while implicitly promoting coordination among subtasks. Experiments on ability-oriented benchmarks show that game-based informal learning improves generalization beyond formal training alone, while CST further enhances multi-task RL by preserving in-domain subtask performance and improving broader general abilities. Code and data are publicly available[1].

1 Introduction

Human intelligence arises from the interplay between formal and informal learning (Scribner and Cole, 1973). Formal learning targets structured, task-specific knowledge, whereas informal learning builds practical wisdom through interaction, experience, and feedback (Callanan et al., 2011). In parallel, recent large language models (LLMs) have achieved remarkable success on formal learning tasks (Liu et al., 2025c; Xu et al., 2025), including mathematical reasoning (Wang et al., 2025b; Zhang et al., 2025b; Chen et al., 2025) and code generation (Seed et al., 2025; Yang et al., 2025; Wang et al., 2025a). However, formal learning alone primarily optimizes LLMs for predefined tasks, motivating us to draw inspiration from human informal learning to develop broader and more generalizable abilities.

This raises a question: what can serve as an informal learning environment for LLMs? We argue that games are a natural choice: they enable learning without explicit instruction, support open-ended interaction through feedback, and abstract diverse real-world scenarios into controllable environments. Building on this insight, we introduce a reinforcement learning framework that jointly trains LLMs on formal mathematical reasoning tasks and informal game-based environments. Specifically, we use three games with increasing interaction complexity: single-turn *Matrix Games*, multi-turn two-player *TicTacToe*, and multi-turn multi-player *Who's the Spy*. Together, they combine formal task structure with informal, experience-driven learning across reasoning, planning, and social interaction.

In this multi-task setting, a straightforward strategy is to mix all subtasks and optimize them with a unified reinforcement learning objective. However, we find that such mixed training suffers from two limitations. At the task-local level, different tasks produce heterogeneous RL signals, including advantages, gradient magnitudes, and exploration dynamics. At the cross-task level, multi-task learning should optimize subtasks jointly while avoiding interference among them. However, mixed training simply follows an aggregated gradient, without explicitly encouraging task-specific gradients to be mutually coordinated.

To address these issues, we propose **Coordinated Subtask Training (CST)**, which replaces a single mixed update with sequential subtask-specific updates. As shown in Fig. 1, the model is

[1] https://anonymous.4open.science/r/GIFT-LLM

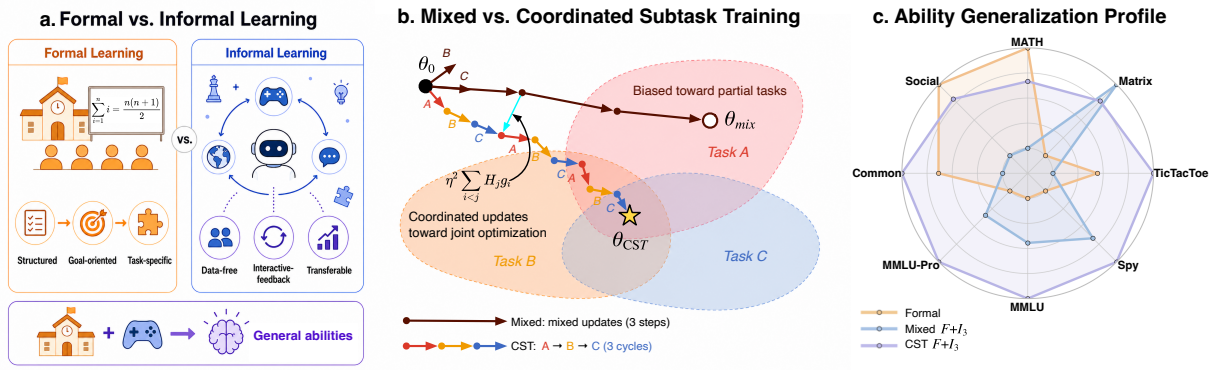

Figure 1: Motivation and framework for informal learning beyond formal task training. (a) We contrast formal and informal learning as two sources of learning signals. (b) We introduce Coordinated Subtask Training (CST), which uses sequential subtask updates to introduce second-order interactions and coordinate multiple subtasks toward a shared objective. (c) 7B-model on the F + I_3 setting results demonstrate improved ability generalization from informal learning and CST.

updated on each subtask in turn, with the resulting parameters passed to the next subtask. This decomposes heterogeneous RL signals into cleaner local optimization steps, yielding more faithful subtask-level policy-improvement directions. Beyond local optimization, CST also improves cross-task coordination. Sequential updates introduce second-order interaction terms $\eta^2 H_j g_i$, which we show are equivalent to encouraging larger pairwise gradient inner products across subtasks. Thus, CST implicitly biases optimization toward regions where subtask gradients are better aligned, so updates reinforce rather than interfere with one another. This helps balance in-domain subtask performance with general capabilities under multi-task RL.

Experimental results on ability-oriented benchmarks show that game-based informal learning improves generalization beyond formal learning alone, increasing the average ability score from 38.34% to 41.13% for 1.5B models and from 42.00% to 57.39% for 7B models. Moreover, CST consistently outperforms mixed training when integrating heterogeneous formal and informal tasks. These results suggest that informal game environments provide useful transferable learning signals, while CST enables them to be absorbed more effectively during multi-task RL training.

Our contributions are summarized as follows:

1) **Game-based informal learning perspective**: We introduce informal learning from educational theory and formulate game as an abstraction of informal learning for LLM training, providing an annotation-free, scalable, and diverse environment;

2) **Methodological Innovation**: We propose **CST**, which replaces coarse mixed updates with sequential subtask-specific updates, preserving task-local RL signals and introducing implicit cross-task gradient coordination.

3) **Empirical Validation**: We show that game-based informal learning improves LLM generalization beyond formal learning alone, while CST better integrates heterogeneous multiple tasks by balancing in-domain subtask performance with broader general abilities.

2 Related Works

2.1 Games and Large Language Models

Recent studies on games and LLMs can be broadly categorized into two directions. The first direction, commonly referred as LLM for Games, investigates the training and evaluation of LLMs within specific game environments. Representative works examine LLM behavior in negotiation games, social deduction settings, and multi-agent text-based games, aiming to assess strategic consistency, equilibrium behavior, or task-specific performance (Fan et al., 2024; Mao et al., 2025; Bianchi et al., 2024; Guertler et al., 2025; Akata et al., 2025). These studies primarily focus on understanding or improving LLM performance within particular games, rather than enhancing general capabilities across tasks. In contrast, the Games for LLM treats games as structured interaction frameworks for improving broader LLM abilities. Prior works demonstrate that self-play, repeated interactions, and multi-agent game dynamics can facilitate

improvements in reasoning, alignment, and strategic adaptation (Tang et al., 2025; Liu et al., 2025b; Xie et al., 2025). However, existing studies remain limited in game diversities and provide insufficient analysis on multi-task settings.

2.2 Multi-Task RL Training in LLMs

Reinforcement learning (RL) has been shown to play a critical role in enhancing the reasoning capabilities of large language models (LLMs) (Liu et al., 2024; Wang et al., 2024; Khatri et al., 2025; Guo et al., 2025; Xu et al., 2025). A growing body of work demonstrates that RL-based training can significantly improve performance on reasoning-intensive tasks, such as mathematical problem solving (Zeng et al., 2025a; Wang et al., 2025b) and code generation(Zhao et al., 2025).More recently, several works have begun to investigate multi-task RL training for LLMs(Zeng et al., 2025b). OMNI-THINKER adopts a curriculum-based training strategy and mixed reward designs (Li et al., 2025) and AgentRL replaces the group-based advantage in GRPO with a task-aware advantage formulation (Zhang et al., 2025a).

In contrast to these approaches, CST offers a simpler optimization-level alternative: it only changes the update structure, separating subtask-local updates and introducing second-order cross-subtask interactions without manually specification.

3 Method

In this section, we explore the motivation of introducing formal and informal learning and how informal learning signals can be integrated into model training through reinforcement learning.

3.1 Motivation

Educational theory distinguishes between *formal* and *informal learning* (Coombs and Ahmed, 1974; Johnson and Majewska, 2022). Formal learning is organized around structured curricula and predefined objectives, whereas informal learning occurs through everyday interaction, practice, and feedback in open-ended environments (Cattell, 1963; Horn and Cattell, 1967). Compared with formal learning, informal learning is often associated with task-independent problem solving and transferable abilities in unseen situations (Ziegler et al., 2012; Thorsen et al., 2014).

Current LLM training largely follows a formal-learning paradigm, where models are optimized on structured tasks with predefined objectives. This improves task-specific performance but may limit the development of transferable capabilities beyond fixed problem domains. Educational research provides further motivation for this direction: informal learning supports problem solving in technology-rich environments (Nygren et al., 2019), contributes substantially to workplace skill development (De Grip, 2024; Fevre et al., 2001), and can be especially beneficial when combined with formal learning (Gerber et al., 2001). Inspired by these findings, we investigate whether interactive, feedback-driven environments can complement formal task-oriented training and improve the generalizability of LLMs.

3.2 Game as Informal Learning Environments

Formal learning emphasizes the structured acquisition of explicit knowledge. In LLM training, this paradigm is naturally instantiated by mathematics, where models solve well-defined problems under supervision. We therefore use *Math* as the formal learning environment in our study.

To model informal learning for LLMs, we ask what can serve as an informal learning environment. We argue that GAME-based environments are a natural choice: they enable learning without explicit instruction, support open-ended interaction through feedback, and abstract diverse real-world scenarios into controllable environments. Moreover, this makes games suitable for experience-driven learning while retaining clear rules and measurable outcomes for reinforcement learning.

Following this view, we select three games with increasing interaction complexity, including single-turn *Matrix Games*,single-turn two-player *TicTacToe*, and multi-turn multi-player *Who's the Spy*, covering different abilities including abstract reasoning, long-horizon planning, and social reasoning. The selected environments are summarized in Table 5, with Details provided in Appendix A.

3.3 From Mixed to Coordinated Subtask Training

To combine formal reasoning tasks with game-based informal learning in LLM training, a natural strategy is to train on a mixture of all subtasks. However, our analysis and experiments show that mixed training introduces limitations at both the task-local and cross-task levels. We therefore propose Coordinated Subtask Training (CST) to address these limitations.

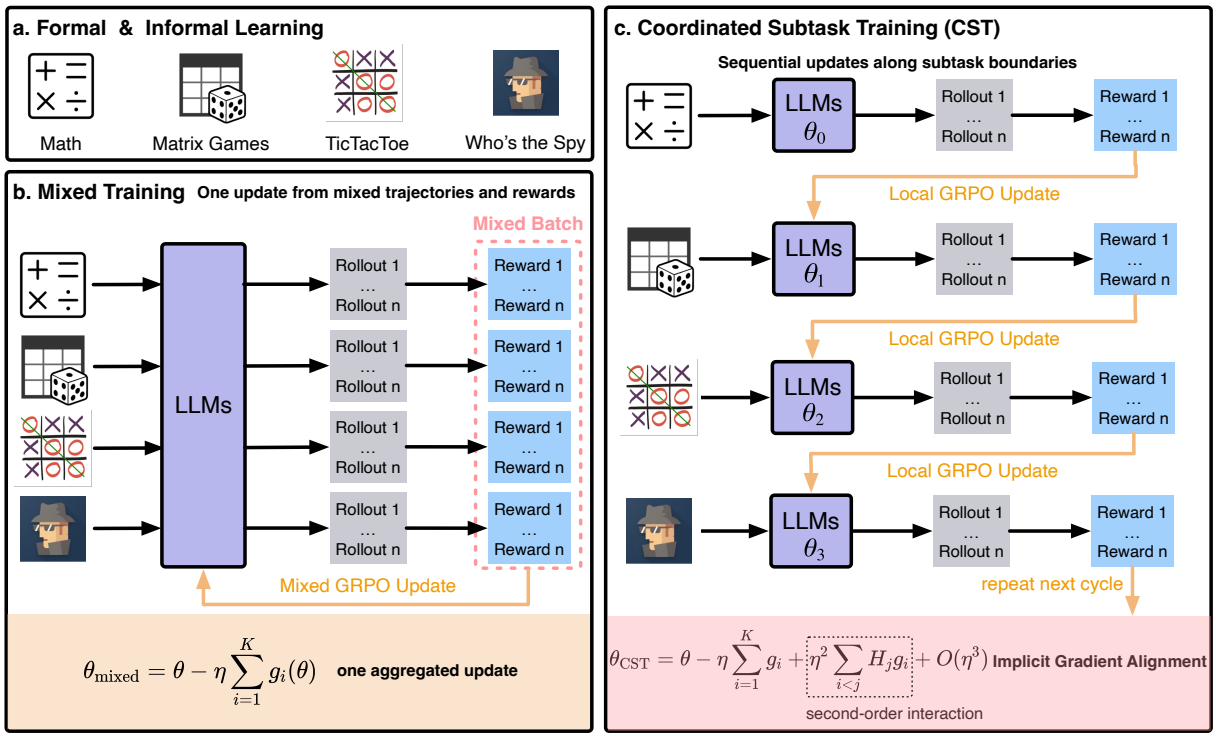

Figure 2: Overview of our motivation, mixed training and CST framework. (a) Formal and Informal learning tasks in our trainings. (b) Mixed training pools rollouts and rewards from all subtasks into a single GRPO update. (c) CST instead follows subtask boundaries: each subtask independently generates rollouts and rewards, performs a local GRPO update, and passes the updated parameters to the next subtask in a cyclic order.

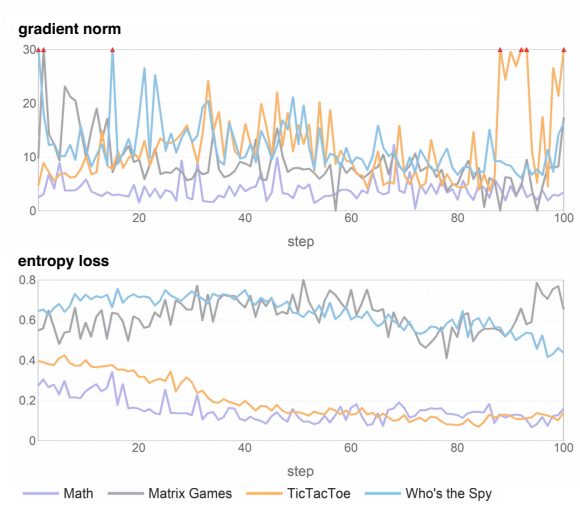

Figure 3: Gradient norms and entropy losses for different subtasks during mixed training.

3.3.1 Mixed Training

In this naive mixed training paradigm, trajectories from different subtasks are placed in the same training batch and optimized through a single policy update. This design exposes the model to diverse environments, but it forces trajectories with different reward structures, exploration dynamics, and learning stages to share the same RL estimation process. We argue that this assumption introduces two optimization limitations: *task-local signal contamination* and *cross-task gradient uncoordination*.

Formally, given K subtasks, the mixed update is wrtten as

$$\theta_{\text{mixed}} = \theta - \eta \sum_{i=1}^{K} g_i(\theta), \tag{1}$$

where $g_i(\theta) = \nabla_\theta \mathcal{L}_{\text{RL}}^{(i)}(\theta)$ denotes the policy gradient estimated from subtask i.

First, mixed training can cause *task-local signal contamination*. The aggregation in Eq. 1 assumes that samples from different subtasks can share the same RL estimation unit. However, subtasks often differ in reward density, exploration demand, trajectory length, and learning stage, leading to distinct gradient scales and entropy patterns, as illustrated in Fig. 3. Under shared batch-level calibration, quantities such as advantages, entropy regularization, and policy-gradient scales are estimated across incompatible subtask statistics rather than within each subtask. As a result, the estimated $g_i(\theta)$ can deviate from a clean subtask-level policy-improvement direction.

Second, mixed training suffers from *cross-task gradient uncoordination*. The direct summation in Eq. 1 treats multi-task learning as a simple aggregation of subtask gradients, but provides no explicit guidance for coordinating their interactions. Although each $g_i(\theta)$ may still contain task-relevant optimization information, the averaged update $\sum_{i=1}^{K} g_i(\theta)$ does not encourage these directions to become mutually compatible. Conflicting components may therefore interfere within the aggregated update, preventing the model from reaching parameter regions that jointly support multiple subtasks and general capabilities.

3.3.2 Coordinated Subtask Training

Therefore, we propose **Coordinated Subtask Training (CST)**, which improves naive mixed training at both the task-local and cross-task levels. CST replaces the single aggregated update in Eq. 1 with a sequence of subtask-specific updates. As shown in Fig. 2, each local update uses trajectories from a single subtask, and the resulting parameters are then passed to the next update. This separates heterogeneous RL signals into homogeneous local estimation units, reducing task-local signal contamination. At the same time, sequentially applying subtask updates allows different optimization directions to interact through parameter transitions. We later show that this interaction induces second-order terms that implicitly encourage gradient compatibility across subtasks, addressing cross-task gradient uncoordination.

For clarity, we first consider the basic CST cycle where each subtask performs one local update before switching to the next subtask:

$$\theta_{\text{CST}} = U_K \circ U_{K-1} \circ \cdots \circ U_1(\theta), \qquad (2)$$

where $U_i(\theta) = \theta - \eta g_i(\theta)$.

A second-order expansion around the initial parameter θ gives

$$\theta_{\text{CST}} = \theta - \eta \sum_{i=1}^{K} g_i + \eta^2 \sum_{i<j} H_j g_i + O(\eta^3), \quad (3)$$

where $g_i = g_i(\theta)$ and $H_i = \nabla_\theta^2 \mathcal{L}_{\text{RL}}^{(i)}(\theta)$ are evaluated at θ. Following the second-order view of Reptile (Nichol et al., 2018), such terms can be connected to pairwise gradient-alignment objectives under an order-averaged approximation:

$$\mathbb{E}_\pi \left[\sum_{i <_\pi j} H_j g_i \right] \approx \frac{1}{2} \nabla_\theta \mathbb{E}_\pi \left[\sum_{i<j} g_i^\top g_j \right], \quad (4)$$

where π denotes the subtask order. Therefore, the second-order term in CST acts like an implicit update toward increasing pairwise gradient inner products across subtasks. Since $g_i^\top g_j = \|g_i\| \|g_j\| \cos(g_i, g_j)$, increasing this quantity encourages smaller angles between subtask gradients when their norms are comparable. This biases optimization toward regions where subtask policy-improvement directions are more compatible, allowing progress on different subtasks to reinforce rather than cancel each other. The detailed derivation and discussion about order-averaging condition are provided in Appendix A.4.

In practice, CST may allow each subtask to perform τ consecutive local updates before switching. This block form is written as

$$\theta_{\text{CST}} = U_K^{(\tau)} \circ U_{K-1}^{(\tau)} \circ \cdots \circ U_1^{(\tau)}(\theta), \qquad (5)$$

where $U_i^{(\tau)}$ denotes applying U_i for τ consecutive steps. The block length τ controls the granularity of CST: larger τ strengthens local consolidation within each homogeneous subtask block, while the same sequential structure preserves the cross-subtask interaction effect above.

4 Experiment

In this section, we present experimental results to evaluate the effectiveness of combining formal and informal learning, as well as the impact of the proposed CST framework.

4.1 Environment Setup

We briefly describe the environments, training configurations, and evaluation protocols used in our experiments, with details in the Appendix.

4.1.1 Environments and Tasks

We adopt *Math* as the formal learning environment and a set of game-based environments as informal learning tasks, including *Matrix Games*, *TicTacToe*, and *Who's the Spy*. For *Math* training, we use MathLv3-5 problems from the SimpleRL-Zoo-Data (Zeng et al., 2025a). For game-based training, we refer to Appendix A for detailed environment descriptions and prompting strategies.

4.1.2 Training Settings

Following the RAGEN framework, we train language models using StarPO*, with trajectory-based reinforcement learning formulation using GRPO. We evaluate two model scales, Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Yang et al.,

Setting	Model	MATH	Matrix	TicTacToe	Spy	MMLU	MMLU-Pro	Common	Social	Avg.
Base	Qwen2.5-1.5B	17.20	7.00	1.00	2.00	37.87	13.49	7.79	27.64	27.10
Formal	*Math*	43.20	21.00	4.00	9.00	51.38	20.97	15.08	65.92	38.34
Informal	*Matrix Games*	19.20	44.00	34.00	14.00	43.79	18.00	9.55	64.12	33.87
	TicTacToe	22.40	32.00	75.00	21.00	47.89	16.45	11.56	64.12	35.00
	Who's the Spy	19.20	21.00	0.00	33.00	43.85	15.68	20.85	60.70	35.27
F + I_1	mixed	37.80	29.00	0.00	16.00	45.19	17.00	12.81	59.11	33.53
	CST	38.00	49.00	11.00	20.00	**52.21**	**22.90**	**16.58**	**64.23**	**38.98**
F + I_2	mixed	28.00	30.00	50.00	20.00	51.35	19.47	17.84	65.30	38.49
	CST	36.80	65.00	41.00	31.00	**52.35**	**20.10**	**24.37**	**67.76**	**41.13**
F + I_3	mixed	35.80	44.00	28.00	17.00	**50.37**	**20.92**	17.09	65.81	38.55
	CST	31.80	55.00	80.00	41.00	49.47	18.39	**28.89**	**66.58**	**40.83**

Table 1: Performance of single-tasks, mixed multi-tasks, and CST multi-tasks training for base model Qwen2.5-1.5B-Instruct, where F + I_k denotes the combination of formal learning and k informal learning tasks. Purple shading highlights the in-domain tasks for each setting.

Setting	Model	MATH	Matrix	TicTacToe	Spy	MMLU	MMLU-Pro	Common	Social	Avg.
Base	Qwen2.5-7B	54.60	40.00	40.00	25.00	71.43	40.17	26.38	75.44	53.36
Formal	*Math*	58.40	39.00	31.00	18.00	66.51	32.45	35.18	75.90	42.00
Informal	*Matrix Games*	58.20	64.00	23.00	27.00	73.22	44.51	26.13	76.20	55.02
	TicTacToe	49.80	57.00	78.00	33.00	73.37	46.35	28.89	76.41	56.26
	Who's the Spy	55.60	43.00	38.00	37.00	71.33	41.17	28.39	74.92	53.95
F + I_1	mixed	58.20	52.00	24.00	21.00	67.55	35.21	27.89	73.34	51.00
	CST	59.40	44.00	32.00	35.00	**68.45**	**39.01**	**35.43**	**74.82**	**54.43**
F + I_2	mixed	50.00	63.00	45.00	9.00	66.80	37.81	36.68	72.21	53.38
	CST	56.20	51.00	55.00	30.00	**68.00**	**35.04**	**46.73**	**74.92**	**56.17**
F + I_3	mixed	56.60	52.00	27.00	34.00	69.24	36.44	31.66	74.62	52.99
	CST	57.80	49.00	36.00	42.00	**72.65**	**44.08**	**37.19**	**75.64**	**57.39**

Table 2: Performance of single-tasks, mixed multi-tasks, and CST multi-tasks training for base model Qwen2.5-7B-Instruct, where F + I_k denotes the combination of formal learning and k informal learning tasks.

2024; Team, 2024), as base models. In two-player and multi-player games, we employ Qwen3-14B as opponents (Team, 2025). For CST, we set the block length to $\tau = 4$ for Qwen2.5-1.5B and $\tau = 1$ for Qwen2.5-7B. We use a larger block length for the smaller model to provide stronger task-local consolidation, while the larger model adopts the basic one-step CST cycle. Full training parameters and implementation are reported in Appendix B.

We report three main experimental settings: single-task training, multi-task mixed training, and multi-task CST. For multi-task configurations, we investigate progressive combinations of formal and informal learning and denote the setups as F + I_k, where k indicates the number of informal learning components included in training, reflecting the depth and complexity of informal learning. Specifically, I_1, I_2, and I_3 correspond to *Matrix Games*, *Matrix Games* combined with *TicTacToe*, and the full combination including *Matrix Games*, *TicTacToe*, and *Who's the Spy*, respectively.

4.1.3 Evaluation Metrics

We evaluate in-domain performance on the training tasks, including the MATH500 benchmark (Lightman et al., 2023), *Matrix Games*, *TicTacToe*, and *Who's the Spy*. For games with opponents, including *TicTacToe* and *Who's the Spy*, we report the average success rate against Gemini-2.5-Flash in 100 rounds (Comanici et al., 2025). For general and diverse abilities, we choose MMLU and MMLU-Pro for multi-domain reasoning (Hendrycks et al., 2021a,b), CommonGen for creative language generation (Lin et al., 2020) and SocialIQA for social reasoning (Sap et al., 2019). In CommonGen tasks, we use GPT-4o (Hurst et al., 2024) to compare the

Setting	Model	MATH	Matrix	TicTacToe	Spy	MMLU	MMLU-Pro	Common	Social	Avg.
F + I_1	TaskAdv	30.40	65.00	10.00	33.00	45.89	18.52	**18.34**	56.55	34.83
	CST	38.00	49.00	11.00	20.00	**52.21**	**22.90**	16.58	**64.23**	**38.98**
F + I_2	TaskAdv	10.40	35.00	46.00	18.00	47.20	17.64	22.11	66.02	38.24
	CST	36.80	65.00	41.00	31.00	**52.35**	**20.10**	**24.37**	**67.76**	**41.13**

Table 3: Comparison between mixed training, task-level normalization, and CST with 1.5B base model.

generated outputs against ground truth references. The details are in Appendix C.

Notation. We abbreviate MATH500, *Matrix Games*, *Who's the Spy*, CommonGen, and SocialIQA as MATH, Matrix, Spy, Common, and Social. Avg. denotes the mean score over MMLU, MMLU-Pro, CommonGen, and SocialIQA. All results are reported as percentages, with the percentage symbol omitted.

4.2 Main Results

Table 1 and 2 summarize the main results across two model scales. We analyze them from two perspectives: the role of formal versus informal learning and the effectiveness of CST method.

Formal vs. Informal Learning. We first examine whether introducing informal learning is empirically meaningful. The results on single-task settings show that formal and informal learning exhibit complementary effects across model scales. For the 1.5B model, formal math training brings most gains on reasoning-oriented benchmarks, indicating the value of structured supervision for smaller models. Meanwhile, informal tasks provide additional benefits by introducing interactive and context-dependent signals. For the 7B model, informal learning yields stronger overall improvements, likely because game-based informal environments introduce less familiar interaction patterns and feedback structures. For instance, *Who's the Spy* promotes contextual description, which aligns with open-ended generation tasks such as CommonGen. These results indicate that formal and informal learning enhance different aspects of model capability.

Effectiveness of CST. Although both formal and informal learning are beneficial when applied in isolation, naively mixing their learning signals does not consistently translate into performance gains. In contrast, CST substantially alleviates performance degradation on general abilites across **all** settings. The effect is particularly pronounced in the F + I_1 setting with 1.5B base model and F + I_3 setting with 7B base model, where CST yields **5.45% and 4.4% absolute improvement**.

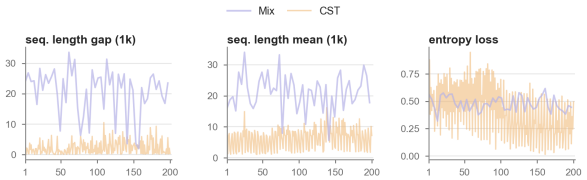

Figure 4: Training dynamics of mixed training and CST in F + I_3 setting with Qwen2.5-7B.

Base Model: Qwen2.5-7B-Instruct MATH500

……a((x-4)^2 - 11) = a(x^2 - 8x + 3)$……
<answer> \\(\\frac{12}{13}\\) </answer>

7B Model Trained on Matrix Games

……Expanding this, we get:\n\\[p(x) = a((x - 4) + \\sqrt{11})((x - 4) - \\sqrt{11}) = a((x - 4)^2 - (\\sqrt{11})^2)……
Since the leading coefficient \\(a\\) can be any integer, we can choose \\(a = 1\\) for simplicity, giving us:
\\[p(x) = x^2 - 8x + 5. \\]……
<answer>\n\\(\\frac{10}{11}\\)\n</answer>

Figure 5: Case Studies on MATH500 benchmark.

To further analyze the effectiveness, we compare the training dynamics of mixed training and CST in Fig. 4, where one mixed step is aligned with four consecutive CST subtask updates. Mixed training shows larger sequence-length gaps and higher average sequence lengths, indicating stronger within-update heterogeneity across subtasks. CST reduces these discrepancies by using subtask-specific local updates, yielding cleaner and more homogeneous RL estimation units. The entropy-loss curve further suggests that CST maintains active exploration dynamics while avoiding the large sequence heterogeneity observed in mixed training.

4.3 Case Study

To illustrate the effectiveness of informal learning, we present a compact case study in Fig. 5 illustrating results of MATH500, with full and detailed comparisons in the Appendix (Fig. 7 and 8). On the MATH500 benchmark, the base 7B model follows the correct high-level reasoning but makes

Setting	Model	MATH	Matrix	TicTacToe	Spy	MMLU	MMLU-Pro	Common	Social	Avg.
F + I_2	mixed	28.00	30.00	50.00	20.00	51.35	19.47	17.84	65.30	38.49
	CST	36.80	65.00	41.00	31.00	**52.35**	**20.10**	**24.37**	**67.76**	**41.13**
I_2	mixed	9.40	49.00	51.00	26.00	39.57	15.78	17.34	59.98	33.17
	CST	10.00	50.00	50.00	37.00	**41.16**	**15.97**	**24.37**	**60.39**	**35.47**

Table 4: Effect of formal learning and CST under the informal-only setting I_2 with 1.5B base model.

a subtle arithmetic error, leading to an incorrect result, whereas the *Matrix Games*-trained model maintains a more explicit and verifiable derivation. Moreover, it chooses $a = 1$ creatively to simplify the calculation. On the CommonGen benchmark, while mixed training on F + I_2 setting produces a reasonable sentence, CST encourages deeper integration by self-validating the result. Together, these cases demonstrate that informal learning with the CST framework promotes more explicit, robust, and creative reasoning across both mathematical and generative tasks.

4.4 Ablation Study

We conduct ablation studies to analyze CST under multi-task RL, comparing it with task-level advantage normalization, assessing the necessity of informal learning, and evaluating its generality across formal-informal task combinations. The analysis of opponent sensitivity is provided in Appendx E.

comparison with Task-Level Advantage. We compare CST with TaskAdv, the task-level advantage normalization strategy proposed in AgentRL (Zhang et al., 2025a). TaskAdv normalizes advantages within each task, partially addressing task-local imbalance, but still optimizes through a single mixed update. In contrast, CST uses separate subtask updates, yielding cleaner local signals and second-order interactions that implicitly promote cross-task gradient alignment. As shown in Table 3, CST achieves a better balance between in-domain subtask performance and broader general abilities than TaskAdv.

Necessity of Formal vs Informal Learning The main experiments already show that combining formal and informal learning yields stronger general abilities than formal learning alone. Here, we further examine whether informal learning is sufficient, as assumed in many game-centric LLM training approaches. Using the same I_2 configuration on the 1.5B base model, we compare informal-only training against the formal+informal(F + I_2) setting. As shown in Table 4, models trained solely on informal learning tasks consistently underperform those trained with both formal and informal learning across most benchmarks, under both mixed training and CST. In particular, removing *Math* signals in CST leads to a performance drop of 11.19% on MMLU and 5.66% on the average general ability score. These results highlight the critical role of formal learning, suggesting that it cannot be fully replaced by informal learning alone.

Generality of CST. To examine whether the CST framework generalizes beyond the combination of formal and informal learning, we evaluate it under an informal-learning-only setting, denoted as I_2, which consists of *Matrix Games* and *TicTacToe*. As shown in Table 4, CST remains effective in this setting, improving CommonGen performance from 17.34% to 24.37% and increasing the average general ability score from 33.17% to 35.47%. These results demonstrate that the CST framework provides stable gains even when applied solely to informal learning environments, indicating its robustness beyond the formal-informal combination.

5 Conclusion

Inspired by theories in cognitive science, we model formal learning as structured *Math* reasoning tasks and informal learning as interactive, game-based environments. We design three representative game settings: *Matrix Games*, *TicTacToe*, and *Who's the Spy*. While a naive mixed training strategy leads to degradation at both task-local and cross-task level, we show that the CST framework enables effective task-local signal isolation and cross-task gradient coordination. Extensive experiments demonstrate that the combination of formal and informal learning is necessary. Compared to mixed training, CST improves generalization across diverse settings consistently, resulting in LLMs with broader abilities.

6 Limitations

This work considers a limited set of formal and informal learning environments, and the game designs represent only a subset of possible interactive settings. While the CST framework generalizes beyond the studied combinations, its effectiveness in more complex or real-world interactive environments remains to be explored. Moreover, our experiments use fixed opponent models in multi-agent games, and extending to self-evolve settings or adaptive opponents is left for future work.

References

Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. 2025. Playing repeated games with large language models. *Nature Human Behaviour*, pages 1–11.

Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. 2024. How well can llms negotiate? negotiationarena platform and analysis. *arXiv preprint arXiv:2402.05863*.

Maureen Callanan, Christi Cervantes, and Molly Loomis. 2011. Informal learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(6):646–655.

Raymond B Cattell. 1963. Theory of fluid and crystallized intelligence: A critical experiment. *Journal of educational psychology*, 54(1):1.

Luoxin Chen, Jinming Gu, Liankai Huang, Wenhao Huang, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Kaijing Ma, and 1 others. 2025. Seed-prover: Deep and broad reasoning for automated theorem proving. *arXiv preprint arXiv:2507.23726*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Philip Hall Coombs and Manzoor Ahmed. 1974. *Attacking rural poverty. How nonformal education can help.*

Andries De Grip. 2024. The importance of informal learning at work. *IZA World of Labor*.

Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. 2024. Can large language models serve as rational players in game theory? a systematic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17960–17967.

Ralph Fevre, S Gorad, and Gareth Rees. 2001. Necessary and unnecessary learning: the acquisition of knowledge and "skills" in and outside employment in south wales in the 20 th century. *The necessity of informal learning*.

Brian L Gerber, Anne ML Cavallo, and Edmund A Marek. 2001. Relationships among informal learning environments, teaching procedures and scientific reasoning ability. *International Journal of Science Education*, 23(5):535–549.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. Textarena. *arXiv preprint arXiv:2504.11442*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

John L Horn and Raymond B Cattell. 1967. Age differences in fluid and crystallized intelligence. *Acta psychologica*, 26:107–129.

Wenyue Hua, Ollie Liu, Lingyao Li, Alfonso Amayuelas, Julie Chen, Lucas Jiang, Mingyu Jin, Lizhou Fan, Fei Sun, William Wang, and 1 others. 2024. Game-theoretic llm: Agent workflow for negotiation games. *arXiv preprint arXiv:2411.05990*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Martin Johnson and Dominika Majewska. 2022. Formal, non-formal, and informal learning: What are they, and how can we research them? research report. *Cambridge University Press & Assessment*.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S Dhillon, David Brandfonbrener, and Rishabh Agarwal. 2025. The art of scaling reinforcement learning compute for llms. *arXiv preprint arXiv:2510.13786*.

Derek Li, Jiaming Zhou, Amirreza Kazemi, Qianyi Sun, Abbas Ghaddar, Liheng Ma, Yu Luo, Dong Li, Jianye HAO, and Yingxue Zhang. 2025. Omni-thinker: Scaling cross-domain generalization in llms

via multi-task rl with hybrid rewards. In *2nd AI for Math Workshop@ ICML 2025*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, and 1 others. 2025b. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*.

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. 2025c. Reinforcement learning meets large language models: A survey of advancements and applications across the llm lifecycle. *arXiv preprint arXiv:2509.16679*.

Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Qiang Guan, Tao Ge, and Furu Wei. 2025. Alympics: Llm agents meet game theory. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2845–2866.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Shota Noda, Kentaro Shirotsuki, and Mutsuhiro Nakao. 2019. The effectiveness of intervention with board games: a systematic review. *BioPsychoSocial medicine*, 13(1):22.

Henrik Nygren, Kari Nissinen, Raija Hämäläinen, and Bram De Wever. 2019. Lifelong learning: Formal, non-formal and informal learning in the context of the use of problem-solving skills in technology-rich environments. *British Journal of Educational Technology*, 50(4):1759–1770.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *EMNLP*.

Sylvia Scribner and Michael Cole. 1973. Cognitive consequences of formal and informal education: New accommodations are needed between school-based learning and learning experiences of everyday life. *Science*, 182(4112):553–559.

ByteDance Seed, Yuyu Zhang, Jing Su, Yifan Sun, Chenguang Xi, Xia Xiao, Shen Zheng, Anxiang Zhang, Kaibo Liu, Daoguang Zan, and 1 others. 2025. Seed-coder: Let the code model curate data for itself. *arXiv preprint arXiv:2506.03524*.

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

Xiaohang Tang, Sangwoong Yoon, Seongho Son, Huizhuo Yuan, Quanquan Gu, and Ilija Bogunovic. 2025. Game-theoretic regularized self-play alignment of large language models. *arXiv preprint arXiv:2503.00030*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Cecilia Thorsen, Jan-Eric Gustafsson, and Christina Cliffordson. 2014. The influence of fluid and crystallized intelligence on the development of knowledge and skills. *British Journal of Educational Psychology*, 84(4):556–570.

Ning Wang, Bingkun Yao, Jie Zhou, Yuchen Hu, Xi Wang, Zhe Jiang, and Nan Guan. 2025a. Large language model for verilog generation with code-structure-guided reinforcement learning. In *2025 IEEE International Conference on LLM-Aided Design (ICLAD)*, pages 164–170. IEEE.

Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu, Guoyin Wang, and Eduard Hovy. 2024. Reinforcement learning enhanced llms: A survey. *arXiv preprint arXiv:2412.10400*.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, and 1 others. 2025b. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025c. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

Yunfei Xie, Yinsong Ma, Shiyi Lan, Alan Yuille, Junfei Xiao, and Chen Wei. 2025. Play to generalize: Learning to reason through game play. *arXiv preprint arXiv:2506.08011*.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jian Yang, Wei Zhang, Shark Liu, Jiajun Wu, Shawn Guo, and Yizhi Li. 2025. From code foundation models to agents and applications: A practical guide to code intelligence. *arXiv preprint arXiv:2511.18538*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025a. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Zhiyuan Zeng, Hamish Ivison, Yiping Wang, Lifan Yuan, Shuyue Stella Li, Zhuorui Ye, Siting Li, Jacqueline He, Runlong Zhou, Tong Chen, and 1 others. 2025b. Rlve: Scaling up reinforcement learning for language models with adaptive verifiable environments. *arXiv preprint arXiv:2511.07317*.

Hanchen Zhang, Xiao Liu, Bowen Lv, Xueqiao Sun, Bohao Jing, Iat Long Iong, Zhenyu Hou, Zehan Qi, Hanyu Lai, Yifan Xu, and 1 others. 2025a. Agentrl: Scaling agentic reinforcement learning with a multi-turn, multi-task framework. *arXiv preprint arXiv:2510.04206*.

Ziyin Zhang, Jiahao Xu, Zhiwei He, Tian Liang, Qiuzhi Liu, Yansi Li, Linfeng Song, Zhenwen Liang, Zhuosheng Zhang, Rui Wang, and 1 others. 2025b. Deeptheorem: Advancing llm reasoning for theorem proving through natural language and reinforcement learning. *arXiv preprint arXiv:2505.23754*.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*.

Matthias Ziegler, Erik Danay, Moritz Heene, Jens Asendorpf, and Markus Bühner. 2012. Openness, fluid intelligence, and crystallized intelligence: Toward an integrative model. *Journal of Research in Personality*, 46(2):173–183.

A Game Rules

We categorize games into three types according to their interaction structure: single-turn games, multi-turn two-player games, and multi-turn multi-player games. For each category, we select a representative game, namely *Matrix Games*, *TicTacToe*, and *Who's the Spy*, respectively. Different games cover different reasoning abilities, as shown in Table. 5.

Following the terminology in VeRL (Sheng et al., 2024), we adopt the concept of **multi-turn** rather than **multi-step**. Here, a **turn** is defined as one complete interaction round in which the trained LLM is queried to produce an action or decision, contributing to the overall game trajectory, while multi-turn refers to a game trajectory with multiple turns. For example, *Who's the Spy* is considered a three-turn game, as each player, as well as the trained LLM, participates in three major interaction rounds: two description turns and one final voting turn. Each turn corresponds to a distinct LLM query, and the sequence of these turns together constitutes a full trajectory.

A.1 Matrix Games

Matrix Games are single-turn strategic reasoning games. We adopt a set of classic matrix games to train LLMs in strategic reasoning, specifically their ability to infer opponents' actions and optimize their own policies toward Nash Equilibrium. In these games, the model must select an action based solely on an abstract payoff table, without access to domain-specific semantics or external knowledge. This setting encourages *abstract reasoning* by requiring the model to interpret symbolic payoffs, compare outcomes across action pairs, and reason about best responses under different opponent choices. At the same time, it fosters *strategic reasoning*: since rewards depend on both players' decisions, the model must anticipate the opponent's likely move, consider mutual incentives, and choose actions that remain robust under strategic interaction, rather than maximizing immediate payoff in isolation. Through reinforcement learning over repeated interactions, the model is trained to align its action selection with equilibrium-consistent behavior, learning to balance self-interest with opponent-aware reasoning.

Category	Game	Targeted Abilities
Single-turn	*Matrix Games*	Abstract & Strategic reasoning;
Multi-turn two-player	*TicTacToe*	Long-horizon planning; Sequential decision making;
Multi-turn multi-player	*Who's the Spy*	Theory of Mind; Creative language generation

Table 5: Representative game environments and the reasoning abilities they promote.

	1	2
1	(3, 3)	(0, 5)
2	(5, 0)	(1, 1)

a. Prisoner's Dilemma

	1	2
1	(2, 1)	(0, 0)
2	(0, 0)	(1, 2)

b. Battle of Sexes

	1	2
1	(-10, -10)	(1, -1)
2	(-1, 1)	(0, 0)

c. Game of Chicken

	1	2
1	(3, 3)	(0, 1)
2	(1, 0)	(1, 1)

d. Stag Hunt

	1	2	3
1	(25, 25)	(50, 30)	(50, 20)
2	(30, 50)	(15, 15)	(30, 20)
3	(20, 50)	(20, 30)	(10, 10)

e. Radio Statio

	1	2	3
1	(13, 3)	(1, 4)	(7, 3)
2	(4, 1)	(3, 3)	(6, 2)
3	(-1, 2)	(2, 3)	(8, -1)

f. IESDS

	1	2
1	(5, 1)	(4, 0)
2	(6, 0)	(3, 1)
3	(6, 4)	(4, 1)

g. Weakly Dominated Game

	1	2	3	4
1	(1, 1)	(-1, 2)	(5, 0)	(1, 1)
2	(2, 3)	(1, 2)	(3, 0)	(5, 1)
3	(1, 1)	(0, 5)	(1, 7)	(0, 1)

h. GAME

	1	2	3	4	5	6
1	(0, 0)	(0, 9)	(0, 14)	(0, 15)	(0, 12)	(0, 5)
2	(9, 0)	(7, 7)	(5, 10)	(3, 9)	(1, 4)	(-1, -5)
3	(14, 0)	(10, 5)	(6, 6)	(2, 3)	(-2, -4)	(-2, -5)
4	(15, 0)	(9, 3)	(3, 2)	(-3, -3)	(-3, -4)	(-3, -5)
5	(12, 0)	(4, 1)	(-4, -2)	(-4, -3)	(-4, -4)	(-4, -5)
6	(5, 0)	(-5, -1)	(-5, -2)	(-5, -3)	(-5, -4)	(-5, -5)

i. Duopolistic Competition

Figure 6: Detailed payoff matrices used in matrix game environments.

Following (Hua et al., 2024), we select a collection of well-studied matrix games that are verified to admit Nash Equilibrium solutions, including Prisoner's Dilemma, Battle of the Sexes, Game of Chicken, Stag Hunt, Radio Station, IESDS, Duopolistic Game, GAME, and Weakly Dominated Game. The corresponding payoff matrices are illustrated in Fig. 6. To prevent the model from overfitting to a fixed numerical scale or specific prompt format, we apply random transformations to the payoff matrices during training. Specifically, each matrix is randomly multiplied by -1 or left unchanged, and an offset of ± 100 is added to all entries. These transformations preserve the strategic structure and equilibrium properties of the games while encouraging the model to focus on relative payoffs and strategic relationships rather than absolute values. In addition, we design multiple instruction prompt templates to present matrix games under diverse linguistic and contextual formulations. This further improves robustness to prompt variations and discourages reliance on superficial patterns. An illustrative example of such a prompt template is shown below:

```
{role}.

Rows = Player 1's actions [{
    p1_actions_list}]; Columns =
    Player 2's actions [{
    p2_actions_list}].

### P1's payoff
{p1_payoff_table}

### P2's payoff
{p2_payoff_table}

{instr}
```

A.2 TicTacToe

TicTacToe is a multi-turn two-player board game in which players alternately place symbols (e.g., X

and O) on a 3×3 grid, with the objective of forming a straight line of three identical symbols horizontally, vertically, or diagonally. Successful play requires each player to anticipate the opponent's potential actions and strategically plan subsequent moves, thereby encouraging the model to reason about others' intentions and future behaviors. In addition, *TicTacToe* imposes strict constraints on valid actions: only unoccupied grid positions can be selected at each turn. These explicit legality requirements provide a clear supervision signal for action validity, which further strengthens the model's instruction-following and rule-compliance capabilities. Prior cognitive and educational studies have also shown that turn-based board games like *TicTacToe* are effective for training planning and strategic reasoning skills (Noda et al., 2019), supporting its suitability as an informal learning environment.

To enhance training robustness and reduce prompt-specific bias, we design four distinct initial prompts that describe the game setting and interaction protocol from different perspectives. Moreover, we observe that the win conditions are not always trivially recognized by LLMs, especially in multi-turn settings. To address this issue, we further introduce an auxiliary win-condition prompt that explicitly explains the winning criteria, including illustrative examples such as horizontal, vertical, and diagonal line completions. During training, the initial prompt is randomly sampled from the prompt set, and the win-condition prompt is also randomly included to improve generalization. Qwen3-14B is employed as the opponent model during training, considering its strong overall capabilities and efficient local inference speed. An illustrative example of the prompt template with the win-condition description is provided below:

```
##Game Rules: TicTacToe
**Objective**: Be the first player to
     connect 3 of your pieces in a
    continuous line.
**Player Pieces**:
- Player 1: 'O'
- Player 2: 'X'
- Empty Slot: '.'
**How to Play**:
1. The game is played on a 3x3
    vertical grid.
2. Players take turns setting one of
    their pieces into any available
    slot.
**Winning Conditions**:
The game ends when a player forms a
    line of 3 of their own pieces. The
     line can be:

1. **Horizontal** (side-by-side in a
    row)
*Example of a horizontal win for
    Player 1 ('O'):*
```
X . .
O O O <-- 3 'O's in row 2
. X .
```
2. **Vertical** (stacked on top of
    each other in a column)
*Example of a vertical win for Player
     2 ('X'):*
```
. X O
O X O <-- 3 'X's in column 2
. X .
```
3. **Diagonal** (connected at an
    angle)
*Example of a diagonal win (bottom-
    left to top-right) for Player 1:*
```
. . O
. O X <-- 3 'O's in a diagonal line
O X .
```
*Example of another diagonal win (top
    -left to bottom-right) for Player
    2:*
```
X . O
. X O <-- 3 'X's in a diagonal line
. O X
```
**Draw Condition**:
If the entire grid is filled with
    pieces and no player has won, the
    game is a draw.

## Current Game State
{state_prompt}

## Your Turn
You are {player}.
The available actions are: {actions}.
```

A.3 Who's the Spy

Who's the Spy is a multi-player, multi-turn social deduction game involving N players, among whom $N-1$ are assigned as *Civilians* and one as the *Undercover*. All players are secretly assigned a word: the civilians share the same word, while the undercover receives a different but semantically related word. Importantly, players do not know anyone's identity throughout the game.

The game proceeds in multiple speaking rounds. In each round, players take turns generating short descriptions of their assigned word, aiming to convey its meaning without revealing too much explicit

information. After two full speaking rounds, all players simultaneously vote to eliminate one suspected undercover. The player receiving the most votes is removed from the game. The civilians win if the undercover is successfully eliminated, while the undercover wins if at least one civilian is voted out.

Winning *Who's the Spy* requires a delicate balance between informativeness and concealment. Civilian players must describe the shared word accurately enough to signal alignment with other civilians, while avoiding overly explicit descriptions that could allow the undercover to infer the civilians' word and adapt accordingly. Conversely, the undercover must generate plausible but strategically ambiguous descriptions to blend in and avoid detection. This interaction demands players to reason about others' beliefs, intentions, and linguistic strategies, thereby strongly engaging theory-of-mind capabilities. At the same time, the open-ended nature of word description encourages flexible and creative language generation.

In our experiments, we fix the number of players to four, consisting of three civilians and one undercover, and set the Qwen3-14B model as training opponents. For each game instance, the identity and speaking order of the trained LLM are randomly assigned. A trajectory is considered successful if the side corresponding to the identity assigned to the trained model wins the game. The word list used in our experiments is adopted from an open-source resource[2]. To improve training robustness and reduce sensitivity to prompt phrasing, we design four distinct rule prompts that describe the game mechanics and player objectives from different perspectives. During training, one rule prompt is randomly sampled for each episode. An example of such a rule prompt is shown below:

```
Game: Who's the Undercover Agent
Roles:
- 3 Civilians share one word.
- 1 Undercover has a related but
    different word.

Goal:
- Civilians: Find the undercover.
- Undercover: Stay hidden until only
    2 players remain.

How to Play:
Each player describes their word in
    one sentence (without saying the
    word itself).
Be subtle yet clear. After two-turn
    speak, everyone votes out one
    player. The one with most votes is
     eliminated.

Win:
- Civilians win if the undercover is
    voted out.
- Undercover wins if one civilian is
    voted out.
```

[2] https://github.com/xzx34/SocialMaze/tree/main/find_the_spy

During the testing phase, we observe that the model tends to repeatedly generate highly similar or identical descriptions across turns, which reduces linguistic diversity and weakens the effectiveness of social deduction training. To mitigate this issue, we introduce an additional hint prompt during the description phase to encourage more varied and informative language generation:

```
### Additional Rules for Description
    (Very Important)
- Your description MUST be clearly
    different from any descriptions
    you have given in earlier rounds.
- Do NOT reuse similar words,
    sentence structures, or ideas.
    Avoid describing it as a "process"
     again.
- Each round, pick a NEW angle of
    interpretation (e.g., its effect,
    its form, its symbolism, its usage
    , etc.)
- The new description should have LOW
     semantic similarity with your
    previous descriptions.
- Act as if you don't remember your
    previous answers, but you MUST
    ensure this answer is not similar
    to them.
- Always produce a new, creative, and
     distinct sentence.
- If the word has multiple meanings,
    assume 100\% that the basic
    meaning is intended. Never choose
    the less common or technical ones.
```

A.4 Derivation of CST Optimization Effect

We provide the detailed derivation of the second-order terms induced by CST. Let $\mathcal{L}_{\mathrm{RL}}^{(i)}(\theta)$ denote the local RL objective for subtask i, and define

$$g_i(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{RL}}^{(i)}(\theta), \qquad H_i(\theta) = \nabla_\theta^2 \mathcal{L}_{\mathrm{RL}}^{(i)}(\theta). \tag{6}$$

For brevity, unless otherwise specified, we write $g_i = g_i(\theta)$ and $H_i = H_i(\theta)$, both evaluated at the initial parameter θ of the CST cycle.

Consider an arbitrary subtask order $\pi = (\pi_1, \ldots, \pi_K)$. A one-step CST cycle is

$$\begin{aligned} \theta_{\text{CST}} &= U_{\pi_K} \circ U_{\pi_{K-1}} \circ \cdots \circ U_{\pi_1}(\theta), \\ U_i(\theta) &= \theta - \eta g_i(\theta). \end{aligned} \tag{7}$$

Let θ_s be the parameter after the first s local updates. Since

$$\theta_{s-1} = \theta - \eta \sum_{r<s} g_{\pi_r} + O(\eta^2), \tag{8}$$

Taylor expansion gives

$$\begin{aligned} g_{\pi_s}(\theta_{s-1}) &= g_{\pi_s}(\theta) + H_{\pi_s}(\theta)(\theta_{s-1} - \theta) + O(\eta^2) \\ &= g_{\pi_s} - \eta H_{\pi_s} \sum_{r<s} g_{\pi_r} + O(\eta^2). \end{aligned} \tag{9}$$

Substituting this into $\theta_s = \theta_{s-1} - \eta g_{\pi_s}(\theta_{s-1})$ and accumulating over $s = 1, \ldots, K$ yields

$$\theta_{\text{CST}} = \theta - \eta \sum_{s=1}^{K} g_{\pi_s} + \eta^2 \sum_{r<s} H_{\pi_s} g_{\pi_r} + O(\eta^3). \tag{10}$$

For the canonical order $(1, \ldots, K)$, this reduces to

$$\theta_{\text{CST}} = \theta - \eta \sum_{i=1}^{K} g_i + \eta^2 \sum_{i<j} H_j g_i + O(\eta^3). \tag{11}$$

The first-order term is identical to the naive mixed update. The difference is the ordered second-order term $\sum_{i<j} H_j g_i$, which appears because later subtask gradients are evaluated at parameters already modified by earlier subtask updates.

We next relate this ordered term to gradient compatibility. Define the pairwise gradient-inner-product objective

$$\mathcal{A}(\theta) = \sum_{i<j} g_i(\theta)^\top g_j(\theta). \tag{12}$$

Assuming the local objectives are twice differentiable, the Hessians are symmetric, and therefore

$$\begin{aligned} \nabla_\theta \mathcal{A}(\theta) &= \sum_{i<j} \nabla_\theta \left(g_i(\theta)^\top g_j(\theta) \right) \\ &= \sum_{i<j} \left(H_i g_j + H_j g_i \right). \end{aligned} \tag{13}$$

Thus, a symmetric combination of the two ordered interactions $H_j g_i$ and $H_i g_j$ corresponds exactly to the gradient of the pairwise inner product.

More generally, let p_{ij} denote the probability, under an order distribution, that subtask i appears before subtask j. Taking expectation over orders in Eq. 10, the ordered interaction term becomes

$$\begin{aligned} \mathbb{E}_\pi \left[\sum_{r<s} H_{\pi_s} g_{\pi_r} \right] = \sum_{i<j} \Big[& p_{ij} H_j g_i \\ & + (1 - p_{ij}) H_i g_j \Big]. \end{aligned} \tag{14}$$

When the order distribution is pairwise balanced, i.e., $p_{ij} = 1/2$ for every pair (i, j), we obtain

$$\mathbb{E}_\pi \left[\sum_{r<s} H_{\pi_s} g_{\pi_r} \right] = \frac{1}{2} \nabla_\theta \mathcal{A}(\theta). \tag{15}$$

This is the same second-order view used by Reptile (Nichol et al., 2018): sequential gradient steps introduce higher-order terms that act like an update toward increasing gradient inner products. Since

$$g_i^\top g_j = \|g_i\| \|g_j\| \cos(g_i, g_j), \tag{16}$$

increasing the inner product encourages smaller angles between subtask gradients when their norms are comparable. This makes the corresponding policy-improvement directions more compatible.

Our implementation uses a fixed cyclic schedule, e.g., $A \to B \to C \to A \to B \to C$. For a single cycle with a fixed starting phase, Eq. 11 is the exact second-order expansion, and the interaction term is directional rather than fully symmetric. Therefore, Eq. 15 should be understood as an order-averaged interpretation, not as an exact identity for one fixed cycle. In continuous cyclic training, however, reverse-direction interactions also occur across cycle boundaries. For a long cyclic sequence containing N complete repetitions, each unordered pair contributes in both directions, and the imbalance between the two directions is only a boundary effect of order $O(N)$ among $O(N^2)$ cross-pair interactions. Thus, the repeated cyclic process provides a practical approximation to the pairwise-balanced view, while still preserving the exact ordered second-order interactions in each local cycle.

We finally extend the derivation to the block form used by CST. Let $U_i^{(\tau)}$ denote applying U_i for τ consecutive local steps. For a single subtask block, Taylor expansion gives

$$U_i^{(\tau)}(\theta) = \theta - \eta \tau g_i + \eta^2 \frac{\tau(\tau - 1)}{2} H_i g_i + O(\eta^3). \tag{17}$$

Applying these blocks sequentially over K subtasks yields

$$\begin{aligned}\theta_{\mathrm{CST}} = \theta - \eta\tau \sum_{i=1}^{K} g_i \\ + \eta^2 \frac{\tau(\tau-1)}{2} \sum_{i=1}^{K} H_i g_i \\ + \eta^2 \tau^2 \sum_{i<j} H_j g_i + O(\eta^3).\end{aligned} \tag{18}$$

The additional within-subtask term reflects local consolidation inside each homogeneous block. The cross-subtask term is the same coordination effect derived above, scaled by τ^2. When $\tau = 1$, Eq. 18 reduces to the basic CST expansion in Eq. 11.

B Detailed Training settings

B.1 Optimization Objective

We adopt a trajectory-based reinforcement learning formulation using Group Relative Policy Optimization (GRPO) as the optimization backbone following the RAGEN framework(Wang et al., 2025c). In multi-turn settings, the model interacts with the environment over a full trajectory τ_i, and a scalar reward is assigned at the trajectory level. The resulting advantage is normalized and distributed across all token positions within the trajectory.

Given a group of G sampled trajectories $\{\tau_i\}_{i=1}^{G}$, we define the policy ratio at token position t as

$$r_{i,t}(\theta) = \frac{\pi_\theta(\tau_{i,(t)} \mid \tau_{i,<t})}{\pi_{\text{old}}(\tau_{i,(t)} \mid \tau_{i,<t})}, \tag{19}$$

and its clipped version as

$$\tilde{r}_{i,t}(\theta) = \text{clip}(r_{i,t}(\theta), 1-\epsilon,\ 1+\epsilon)\,. \tag{20}$$

The trajectory-level GRPO objective is then given by

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\min\left[r_{i,t}(\theta)\hat{A}_{i,t},\ \tilde{r}_{i,t}(\theta)\hat{A}_{i,t}\right], \tag{21}$$

where $\hat{A}_{i,t}$ denotes the normalized advantage at token position t within trajectory τ_i, and ϵ is the clipping threshold.

This objective follows a standard PPO-style formulation and serves as a unified optimization backbone for all training settings in this work.

B.2 Coordinated Subtask Training Framework

Coordinated Subtask Training (CST) reformulates multi-task RL by sequentially applying subtask-specific policy updates instead of optimizing all subtasks through a single mixed update. Specifically, a CST cycle consists of an ordered sequence of local updates, where each update uses trajectories sampled from one subtask only. Each subtask preserves its original interaction protocol, reward function, and success condition, while the model parameters are passed from one local update to the next.

This formulation differs from naive mixed training. In mixed training, trajectories from different subtasks are placed in the same batch and optimized through one aggregated policy update. This can contaminate task-local RL signals because advantages, entropy terms, and policy-gradient scales are calibrated over heterogeneous trajectory distributions. In contrast, CST estimates each local gradient within a homogeneous subtask-specific batch, yielding cleaner task-local optimization signals.

CST also changes how subtasks interact during optimization. Instead of directly summing gradients from different subtasks, CST applies them sequentially. Therefore, later subtask updates are computed on parameters already modified by earlier subtasks. This sequential structure introduces additional second-order cross-subtask terms, such as $\eta^2 H_j g_i$ for two subtasks and $\eta^2(H_B g_A + H_C g_A + H_C g_B)$ for three subtasks ordered as $A \to B \to C$. These terms are absent from direct gradient summation and provide an implicit mechanism for coordinating subtask optimization directions.

In practice, CST can be implemented with a block length τ, where each subtask performs τ consecutive local updates before switching to the next subtask. The case $\tau = 1$ recovers the basic CST cycle, while larger τ strengthens local consolidation within each homogeneous subtask block. Moreover, we use a fixed cyclic schedule over subtasks, e.g., $A \to B \to C \to A \to B \to C$, so that all subtasks are repeatedly optimized while maintaining clear subtask boundaries.

B.3 Training Configuration and Implementation

Following the RAGEN framework based on VeRL (Sheng et al., 2024), we utilize the stable StarPO* algorithm. The key improvements in StarPO* are DAPO (Yu et al., 2025) and trajectory filtering. DAPO removes the KL-term and employs a clip-higher strategy. The trajectory filtering is based on variance, where only the top 25% of trajectories with the highest variance are retained for each round.

Regarding hyperparameters, each setting runs a maximum of 250 rollout-upodate iterations, with the group size fixed to 16. During training, we fixed the random seed to generate the task prompts to ensure reproducibility. We use $\tau = 4$ for Qwen2.5-1.5B and $\tau = 1$ for Qwen2.5-7B as fixed training settings across all CST experiments for each model scale. Exploring adaptive or task-dependent block lengths is left for future work.

The maximum number of turns for each subtask is determined by the nature of the task. For single-turn tasks such as *Math* and *Matrix Games*, the maximum number of turns is 1. For multi-turn tasks, the maximum turns for *TicTacToe* is set to 5, since the game involves a 3x3 grid, and the game will always end after the 5th turn. In *Who's the Spy*, the maximum turns is 3, consisting of two description rounds and one voting round. In mixed training and CST, the `max_turn` is set to the maximum number of turns across all sub-tasks.

The model is optimized using Generalized Advantage Estimation (GAE) with $\gamma = 1.0$ and $\lambda = 1.0$, along with the Adam optimizer where $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We also apply entropy regularization with a coefficient $\beta = 0.001$. The reward is defined as 1 for success and 0 for failure, with a reward of 0.5 assigned to draws in *TicTacToe*. The format penalty of -0.1 is applied.

In multi-task settings, both mixed training and CST scenarios involve an equal number of tasks, ensuring fairness in task comparison.

C Detailed Evaluation Settings

We introduce the detailed evaluation settings including math, game and general abilities scenarios.

C.1 Math Evaluation

We evaluate the model's math reasoning ability using the MATH500 benchmark. Model outputs are compared against ground-truth answers using the Math-Verify toolkit[3], which provides a robust verification pipeline for mathematical equivalence and correctness.

C.2 Game Evaluation

To assess game-playing ability, we evaluate the model's success rate over 100 independent game rounds, with the opponent model fixed to Gemini-2.5-Flash for *TicTacToe* and *Who's the Spy* tasks. For each game, the success criterion is defined in accordance with the corresponding training objective. Specifically, in *Matrix Games*, a trajectory is considered successful if the selected action satisfies the Nash Equilibrium condition. In *TicTacToe*, success is defined as achieving a valid three-in-a-row configuration (horizontal, vertical, or diagonal) for the trained model's pieces. To make the evaluation of *TicTacToe* fair, models are tested on an empty board and the first player is randomly selected. In *Who's the Spy*, a trajectory is counted as successful if the side corresponding to the trained model's assigned identity wins the game.

C.3 General Abilities Evaluation

To evaluate the model's generalization across diverse ability dimensions, we select a set of widely used benchmarks, including MMLU, MMLU-Pro, CommonGen, and SocialIQA.

MMLU and MMLU-Pro. MMLU covers a broad range of academic subjects and evaluates knowledge and reasoning through four-choice questions. MMLU-Pro extends MMLU by increasing both task difficulty and answer space, expanding each question from four options to ten options, thereby posing a more challenging evaluation setting.

CommonGen. The CommonGen benchmark is designed to evaluate the model's language generation ability by requiring it to produce a coherent and plausible sentence that incorporates a given set of concept words with specified semantic roles, as illustrated in Fig. 8. Since CommonGen is an open-ended generation task, we follow the evaluation protocol provided by allenai[4], using few-shot prompting and GPT-4o as an automatic evaluator to compare model outputs against human-annotated reference sentences. To reduce positional bias during judgment, we randomly permute the order of

[3] https://github.com/huggingface/Math-Verify
[4] https://github.com/allenai/CommonGen-Eval

the model-generated output and the reference sentence, and instruct the evaluator to carefully compare both candidates. Since the human-annotated references represent a strong upper bound that is difficult to consistently surpass, we report the success rate using a win-or-tie criterion. The detailed evaluation prompt is provided below:

```
# Data
Given several concepts (i.e., nouns
    or verbs), we ask models to write
    a short and simple sentence that
    contains *all* the required words.
The sentence should describe a common
     scene in daily life, and the
    concepts should be used in a
    natural way.
Concepts: "{concept_list}"
Model A: "{candidate_A}"
Model B: "{candidate_B}"

# Your Task
Your task is to choose a better
    sentence from the two candidates.
    Decide which model's sentence is
    better in terms of the naturalness
     and commonness of the scenes they
     describe.

## Rules:
- A better sentence should describe a
     common scene in daily life, and
    all concepts should be used in a
    natural way.
- You should prefer sentences that
    use all given concepts with
    correct part-of-speech tags.
- A simpler and shorter sentence is
    preferred if it describes the same
     scene as the other sentence.
- If you think both sentences are
    equally good or bad, please choose
     "tie".

Now, please output your choice ("A"
    or "B" or "tie").

Your choice:
```

SocialIQA. SocialIQA is a question-answering benchmark designed to evaluate social reasoning through three-choice questions. Unlike prior benchmarks that emphasize physical or taxonomic knowledge, SocialIQA focuses on understanding people's actions and their social motivations. Given a described action, the model must infer the most plausible social intent or implication among multiple candidates. The dataset covers a wide range of everyday social situations, with answer options consisting of both human-written and adversarially filtered machine-generated candidates.

D Detailed Case Studies

Effectiveness of Informal Learning on Mathematical Reasoning. To provide a more detailed illustration of how informal learning improves mathematical reasoning, we present a representative case study on the MATH500 benchmark in Fig. 7. The example compares the base *Qwen2.5-7B-Instruct* model with the same model after training on *Matrix Games*.

In this example, both models correctly identify that if a quadratic polynomial with integer coefficients has $4-\sqrt{11}$ as a root, then its conjugate $4+\sqrt{11}$ must also be a root. The base model further recognizes the high-level structure of the solution and attempts to construct the corresponding polynomial. However, it commits a subtle arithmetic error when expanding $(x-4)^2-11$, incorrectly computing $16-11$ as 3. This local mistake propagates to the subsequent evaluation of $p(3)$ and $p(4)$, ultimately leading to an incorrect final answer, despite the overall reasoning strategy being correct.

In contrast, the model trained with *Matrix Games* exhibits a more structured and explicit derivation. Instead of directly simplifying intermediate expressions, it expands the polynomial step by step as

$$(x-4)^2-11 = x^2-8x+16-11,$$

making the critical arithmetic operation transparent and easy to verify. Moreover, it explicitly observes that the leading coefficient a can be any integer and chooses $a=1$ for simplicity, demonstrating greater flexibility and creativity in solution construction. This explicit and verifiable reasoning process helps the model avoid local arithmetic errors and arrive at the correct final result. This case suggests that informal learning through interactive game environments encourages more cautious, structured, and self-checking reasoning behaviors in mathematical problem solving.

Effectiveness of CST on Semantic Generation. We further analyze the difference between mixed and CST strategies through a detailed case study on the CommonGen benchmark, as shown in Fig. 8. The task requires generating a coherent daily-life sentence that naturally incorporates all given concepts.

The model trained with mixed objectives is able to correctly include the required concepts and produce a grammatically valid sentence. However, its generation primarily focuses on satisfying individ-

Setting	Opponent	MATH	Matrix	TicTacToe	Spy	MMLU	MMLU-Pro	Common	Social	Avg.
TicTacToe	Gemini-2.5-Flash	25.80	17.00	46.00	12.00	39.00	13.79	14.32	51.18	29.57
	Qwen3-14B	22.40	32.00	75.00	21.00	47.89	16.45	11.56	64.12	35.00
Who's the Spy	Gemini-2.5-Flash	19.20	19.00	2.00	38.00	46.86	17.21	20.85	59.57	36.12
	Qwen3-14B	19.20	21.00	0.00	33.00	43.85	15.68	20.85	60.70	35.27

Table 6: Opponent sensitivity analysis of *TicTacToe* and *Who's the Spy* training for base 1.5B model, comparing Gemini-2.5-Flash and Qwen3-14B as opponents.

Question: Let $p(x)$ be a quadratic polynomial with integer coefficients which has $4-\\sqrt{11}$ as a root. Compute $\\frac{p(3)}{p(4)}.$

Base Model: Qwen2.5-7B-Instruct

<think> Since $p(x)$ is a quadratic polynomial with integer coefficients and $4-\\sqrt{11}$ is a root, its conjugate $4+\\sqrt{11}$ must also be a root. This is because the coefficients are integers, and the polynomial must factor over the reals to maintain integer coefficients. The polynomial can thus be written as $p(x) = a(x - (4-\\sqrt{11}))(x - (4+\\sqrt{11})) = a((x-4)^2 - 11) = a(x^2 - 8x + 3)$. We need to find $\\frac{p(3)}{p(4)}$. First, calculate $p(3)$ and $p(4)$.
For $p(3)$:\n\\[p(3) = a(3^2 - 8\\cdot3 + 3) = a(9 - 24 + 3) = a(-12)\\]
For $p(4)$:\n\\[p(4) = a(4^2 - 8\\cdot4 + 3) = a(16 - 32 + 3) = a(-13)\\]
Thus, $\\frac{p(3)}{p(4)} = \\frac{-12a}{-13a} = \\frac{12}{13}$. The value of a cancels out, so the result is independent of a.</think>
<answer> \\(\\frac{12}{13}\\) </answer>

Model Trained on Matrix Games

<think>\nSince \\(p(x)\\) is a quadratic polynomial with integer coefficients and \\(4 - \\sqrt{11}\\) is a root, its conjugate \\(4 + \\sqrt{11}\\) must also be a root. This is because the coefficients are integers, and the polynomial must have roots that come in conjugate pairs if they are irrational.
Let \\(p(x) = a(x - (4 - \\sqrt{11}))(x - (4 + \\sqrt{11}))\\). Expanding this, we get:\n\\[p(x) = a((x - 4) + \\sqrt{11})((x - 4) - \\sqrt{11}) = a((x - 4)^2 - (\\sqrt{11})^2) = a(x^2 - 8x + 16 - 11) = a(x^2 - 8x + 5). \\]
Since the leading coefficient \\(a\\) can be any integer, we can choose \\(a = 1\\) for simplicity, giving us:
\\[p(x) = x^2 - 8x + 5. \\]
Now, we need to compute \\(\\frac{p(3)}{p(4)}\\):
\\[p(3) = 3^2 - 8 \\cdot 3 + 5 = 9 - 24 + 5 = -10, \\]
\\[p(4) = 4^2 - 8 \\cdot 4 + 5 = 16 - 32 + 5 = -11. \\]
Thus, \\(\\frac{p(3)}{p(4)} = \\frac{-10}{-11} = \\frac{10}{11}\\).
The final answer is \\(\\frac{10}{11}\\).</think>
<answer>\n\\(\\frac{10}{11}\\)\n</answer>

Ground Truth: \\frac{10}{11}

Figure 7: Detailed case study on MATH500, comparing 7B base model trained on Math with *Matrix Games*.

ual lexical constraints, resulting in a relatively minimal and less vivid description. While the output is semantically acceptable, it lacks explicit scene construction and deeper integration among the concepts.

In contrast, the model trained with CST demonstrates a more deliberate generation process. Before producing the final sentence, it briefly verifies the required concepts and analysis its answer. This leads to a more expressive, contextually rich, and human-aligned sentence. As a result, the CST model achieves a win-or-tie outcome when compared against strong human-annotated references.

This comparison highlights that CST encourages deeper semantic integration and scene-level reasoning, going beyond word-level constraint satisfaction. By requiring the model to jointly consider multiple objectives within a single trajectory, CST promotes more coherent and expressive language generation than naive mixed training.

E Detailed Ablation Studies

E.1 Opponent Sensitivity in Multi-player Games

We study the effect of opponent choice by training the 1.5B model against different opponents, as shown in Table 6. Overall, the choice of opponent mainly affects the game on which the model is trained, while its impact on general benchmark performance is relatively small.

For *TicTacToe* training, using Qwen3-14B as the opponent substantially improves the *TicTacToe* success rate from 46% to 75%, yielding a 29-point absolute gain over Gemini-2.5-Flash. This suggests that training against the stronger Qwen3-14B opponent encourages the model to learn more robust game-specific strategies. In addition, the average score increases from 29.57 to 35.00, indicating that the stronger opponent does not degrade overall capability. Although performance on some non-game benchmarks slightly decreases, such as MATH from 25.80 to 22.40 and Common from 14.32 to 11.56, the model obtains higher scores on Matrix, MMLU, MMLU-Pro, and Social, showing no clear evidence of broad negative transfer.

Instruction
Given several concepts (i.e., nouns or verbs), write a short and simple sentence that contains *all* the required words.
The sentence should describe a common scene in daily life, and the concepts should be used in a natural way.
Examples
Example 1
- Concepts: \"dog(noun), frisbee(noun), catch(verb), throw(verb)\"
- Sentence: The dog catches the frisbee when the boy throws it into the air.
Example 2
- Concepts: \"apple(noun), place(verb), tree(noun), pick(verb)\"
- Sentence: A girl picks some apples from a tree and places them into her basket.
Your Task
- Concepts: \"food(noun), front(noun), sit(verb), table(noun)\"

7B Model after Mixed Training

<think>The sentence needs to include all the given concepts in a natural daily life scenario. \"Food\" can be something placed on a table, \"front\" can refer to the front of the table, and \"sit\" can describe the action of someone sitting down.</think> **List concepts.**

Ground Truth:
The boy sat at the front of the table and waited for his food.

7B Model after CST

<think>Given the concepts: "food (noun), front (noun), sit (verb), table (noun)", we need to create a common daily life scene.\n\n1. Identify a typical setting: A dining area with a table.\n2. Involving "sit (verb)": People usually sit at a table.\n3. Incorporate "food (noun)": Food is commonly eaten at a table.\n4. Use "front (noun)": We can describe a focal point or position …… **List concepts.**

This sentence naturally includes all required words in a common daily life scenario:\n- "sits" (sit) describes the action of the family members.\n- "table" (table) is the common setting for dining.\n- "main course" (food) is the focal item being eaten.\n- "in front of" (front) describes the position of the food relative to each person.\n\nThus, the sentence effectively uses all given concepts in a natural and common daily life context.</think> **Analysis**

Figure 8: Detailed case study on CommonGen, comparing 7B base model on F+I_2 settings mixed training with CST.

Model	MATH	Matrix	TicTacToe	Spy	MMLU	MMLU-Pro	Common	Social	Avg.
GPT-4o	59.80	48.00	63.00	70.00	85.08	66.00	53.02	79.02	70.78
Gemini3-Flash	49.80	64.00	86.00	16.00	78.42	52.01	41.96	74.87	61.82
DeepSeek-V3.2	55.80	61.00	88.00	69.00	86.73	63.69	67.34	80.19	74.49

Table 7: Additional results of close-source models.

For *Who's the Spy* training, the difference between opponents is smaller. Training against Gemini-2.5-Flash achieves a higher *Who's the Spy* success rate of 38%, compared with 33% when training against Qwen3-14B. However, Qwen3-14B still provides competitive overall performance, with an average score of 35.27 compared with 36.12 for Gemini-2.5-Flash. The two settings also show similar performance on most general benchmarks, suggesting that opponent choice has a weaker effect in *Who's the Spy* than in *TicTacToe*.

Notably, Qwen3-14B outperforms Gemini-2.5-Flash in direct evaluations, achieving success rates of 66% in *TicTacToe* and 52% in *Who's the Spy*. Together with its strong training performance in *TicTacToe* and the practical advantage that the open-sourced Qwen3-14B can be deployed locally with lower inference latency, we adopt Qwen3-14B as the default opponent in all experiments.

F Additional Results of Closed-Source models

We further evaluate a set of strong closed-source models, including GPT-4o, Gemini-3-Flash-Preview, and DeepSeek-V3.2 (chat mode, without reasoning) (Liu et al., 2025a). All evaluation protocols are strictly aligned with those used in the main experiments. The corresponding results are reported in Table 7.

Overall, these results demonstrate that strong closed-source models exhibit competitive performance across both formal and informal learning benchmarks, while showing distinct strengths on different task categories. GPT-4o achieves balanced performance across mathematical reasoning, social reasoning, and general knowledge benchmarks, reflecting its strong general-purpose capability. Gemini-3-Flash-Preview performs particularly well on interactive game tasks such as *TicTacToe* and *Matrix Games*, but shows relatively weaker performance on social deduction and general language generation tasks. DeepSeek-V3.2 achieves

the highest overall average score, with consistently strong results on math, game, and general ability benchmarks, indicating that competitive generalization can be achieved even without explicit reasoning traces.These observations suggest that while closed-source models benefit from large-scale training and strong base capabilities, their performance profiles vary substantially across different ability dimensions.

Moreover, we observe that the 1.5B and 7B models after RL training can, under certain domain-specific or CST settings, achieve performance comparable to or even surpass strong closed-source models on targeted benchmarks. As shown in the main results, a 1.5B model trained under the $F + I_1$ setting attains 65.00% accuracy on *Matrix Games*, exceeding the performance of all evaluated closed-source models in this domain.

Similar phenomena are also observed for 7B models on several general ability benchmarks, including MMLU-Pro, CommonGen, and SocialIQA, where models trained with CST or domain-aligned tasks achieve results on par with those of closed-source APIs. These gains are particularly pronounced on tasks that require structured reasoning, interaction, or social inference, which are directly targeted by the corresponding informal learning environments.

Taken together, these results indicate that, despite the substantially smaller model size and more limited training data, our proposed training framework can effectively elicit strong domain-specific and transferable capabilities through structured and CST reinforcement learning. This comparison with closed-source models provides further evidence that carefully designed interactive training objectives can partially compensate for scale and serve as an efficient alternative for capability enhancement.