STRATAGEM: Learning Transferable Reasoning via Trajectory-Modulated Game Self-Play

Anonymous ACL submission

Abstract

Games offer a compelling paradigm for developing general reasoning capabilities in language models, as they naturally demand strategic planning, probabilistic inference, and adaptive decision-making. However, existing self-play approaches rely solely on terminal game outcomes, providing no mechanism to distinguish transferable reasoning patterns from game-specific heuristics. We present STRATAGEM, which addresses two fundamental barriers to reasoning transfer: *domain specificity*, where learned patterns remain anchored in game semantics, and *contextual stasis*, where static game contexts fail to cultivate progressive reasoning. STRATAGEM selectively reinforces trajectories exhibiting abstract, domain-agnostic reasoning through a *Reasoning Transferability Coefficient*, while incentivizing adaptive reasoning development via a *Reasoning Evolution Reward*. Experiments across mathematical reasoning, general reasoning, and code generation benchmarks demonstrate substantial improvements, with particularly strong gains on competition-level mathematics where multi-step reasoning is critical. Ablation studies and human evaluation confirm that both components contribute to transferable reasoning.[1]

1 Introduction

Games have long served as a proving ground for artificial intelligence, offering structured environments where complex reasoning emerges from simple rules (Silver et al., 2016; Berner et al., 2019; Vinyals et al., 2019). Beyond serving as evaluation benchmarks, games provide a unique opportunity for cultivating general reasoning capabilities: they demand strategic planning, probabilistic inference, and adaptive decision-making, all cognitive skills that underpin intelligent behavior across diverse domains (Xu et al., 2024; Hu et al., 2025). This observation has motivated a growing body of work exploring games as training environments for language models (Hu et al., 2024; Tong et al., 2025; Xie et al., 2025), premised on the hypothesis that reasoning patterns developed through gameplay may transfer to downstream tasks such as mathematical problem-solving and code generation.

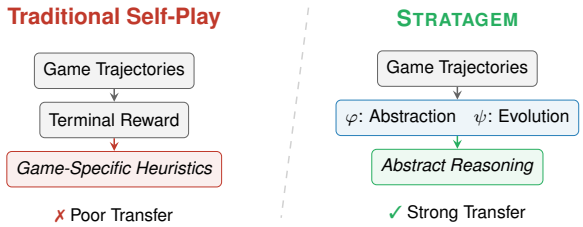

Figure 1: Traditional self-play learns game-specific heuristics from terminal rewards. STRATAGEM modulates trajectory advantages via abstraction (φ) and evolution (ψ), selectively reinforcing transferable reasoning.

Self-play has emerged as a promising paradigm within this agenda, enabling models to improve through competitive interaction without requiring curated datasets (Zhang et al., 2024; Zhao et al., 2025). Historical successes in game-playing AI, from AlphaGo (Silver et al., 2016) to OpenAI Five (Berner et al., 2019), demonstrate that self-play can produce superhuman performance in specific domains. Recent work has extended this paradigm to language models: SPIRAL (Liu et al., 2025) trains LLMs through self-play on text-based zero-sum games, showing that game-derived rewards can improve reasoning capabilities. However, SPIRAL relies on terminal game outcomes (win/loss) to provide learning signals, offering no explicit mechanism to identify or reinforce reasoning patterns that transfer beyond game-specific contexts. As a result, models may learn to win games through domain-specific heuristics (e.g., "King beats Queen") that fail to generalize, while transferable reasoning (e.g., "enumerate cases and compute expected value") receives no preferential reinforcement.

[1] Code and models will be released upon publication.

To address this limitation, we propose STRATAGEM (**S**elf-Play **TR**ajectory **A**dvan**T**age **A**ctivated **G**am**E** Lear**M**ing), which learns transferable reasoning by selectively reinforcing trajectories that exhibit domain-agnostic and adaptive reasoning patterns. Our key insight is that transfer requires addressing two fundamental challenges: *domain specificity*, where game-learned patterns remain anchored in game semantics rather than abstract principles; and *contextual stasis*, where static game environments fail to cultivate the progressive reasoning needed for evolving problem contexts. STRATAGEM tackles both challenges by modulating trajectory advantages with two complementary signals: a *Reasoning Transferability Coefficient* (φ) that measures the abstraction level of reasoning patterns, and a *Reasoning Evolution Reward* (ψ) that incentivizes reasoning that deepens and adapts across turns. By multiplicatively scaling advantage based on transferability and additively rewarding reasoning evolution, STRATAGEM ensures that only trajectories demonstrating both abstract reasoning and progressive development receive maximal reinforcement.

We evaluate STRATAGEM on benchmarks spanning mathematical reasoning, general reasoning, and code generation. Training on three text-based games using Qwen3-4B-Base, STRATAGEM achieves consistent improvements across all categories, with strong gains on competition-level mathematics where multi-step reasoning is critical. Ablation studies confirm that both modulation components contribute meaningfully, while human evaluation validates that STRATAGEM produces more abstract and progressive reasoning.

Our contributions are:

- We identify *domain specificity* and *contextual stasis* as two fundamental barriers to reasoning transfer in game-based self-play, and propose STRATAGEM to address both through selective trajectory advantage modulation.
- We introduce the Reasoning Transferability Coefficient (φ) that quantifies abstraction level, and Reasoning Evolution Reward (ψ) that incentivizes progressive reasoning development.
- We demonstrate strong transfer across mathematical reasoning, general reasoning, and code generation, with notable gains on competition-level problems requiring multi-step reasoning.

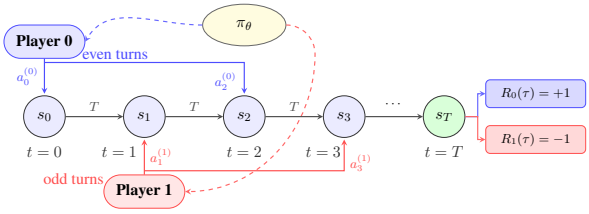

Figure 2: Two-player zero-sum Markov game structure. Both players share a single policy π_θ with role conditioning. Players alternate turns: Player 0 acts at even timesteps ($t \bmod 2 = 0$), Player 1 at odd timesteps. The transition function T governs state dynamics based on actions. At terminal state s_T, rewards satisfy the zero-sum constraint $R_0(\tau) + R_1(\tau) = 0$.

2 Preliminaries

2.1 Task Formulation

We formulate multi-turn reasoning as a turn-level Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \gamma)$, where states $s \in \mathcal{S}$ represent complete contexts (e.g., game configurations) and actions $a \in \mathcal{A}$ correspond to full responses rather than individual tokens (see Appendix G for extended background). At each turn t, the model generates response y_t containing reasoning c_t and executable action a_t.

For competitive interactions, we extend this to a two-player zero-sum Markov game (Littman, 1994) $\mathcal{G} = (\mathcal{S}, \mathcal{A}_0, \mathcal{A}_1, T, r, \gamma)$ with opposed rewards:

$$r_0 + r_1 = 0 \;\; \forall (s, a^{(0)}, a^{(1)}), \quad R_1(\tau) = -R_0(\tau). \tag{1}$$

Figure 2 illustrates this structure for trajectory $\tau = \{(s_t, a_t^{(0)}, a_t^{(1)})\}_{t=0}^{T}$.

2.2 SPIRAL

SPIRAL (Liu et al., 2025) trains language models through self-play on turn-based zero-sum games $\mathcal{G} = \{G_1, \ldots, G_n\}$ with sparse terminal rewards $R_p(\tau) \in \{-1, 0, 1\}$ (see Appendix K for details). Both players share a single policy π_θ with role conditioning: player $p = t \bmod 2$ generates $y_t^{(p)} \sim \pi_\theta(\cdot|s_t, p, G)$ at turn t.

To handle asymmetric expected returns across roles, SPIRAL employs Role-conditioned Advantage Estimation (RAE) with separate baselines $b_{G,p}$ per game-role pair:

$$A_{G,p}(\tau) = R_p(\tau) - b_{G,p}, \tag{2}$$
$$\nabla_\theta J = \mathbb{E}\Big[\textstyle\sum_{t \in \mathcal{T}_p} A_{G,p} \nabla_\theta \log \pi_\theta(y_t^{(p)}|s_t)\Big],$$

where $\mathcal{T}_p$ indexes turns of player p and baselines are updated via exponential moving average.

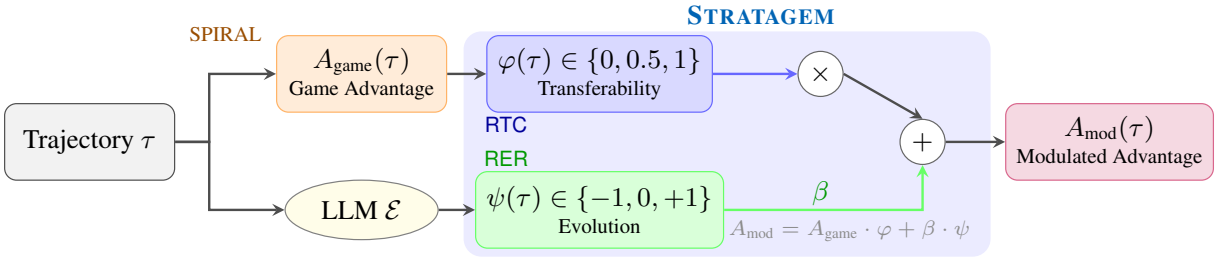

Figure 3: Overview of STRATAGEM. Given a trajectory τ from self-play, the game-based advantage A_{game} is computed. STRATAGEM modulates this advantage using two signals: the Reasoning Transferability Coefficient φ that multiplicatively scales the advantage based on cross-domain transfer potential, and the Reasoning Evolution Reward ψ that additively rewards reasoning development within trajectories.

3 Method

This section presents STRATAGEM, which selectively reinforces transferable reasoning patterns through trajectory advantage modulation. We first provide an overview (§3.1), then detail the Reasoning Transferability Coefficient (§3.2) and Reasoning Evolution Reward (§3.3).

3.1 Overview

Transferring reasoning capabilities from games to domains such as mathematics and coding faces two fundamental challenges:

1. **Domain Specificity**: Reasoning patterns learned from games tend to be anchored in game-specific concepts, terminology, and heuristics (e.g., "King beats Queen") rather than abstract, domain-agnostic patterns (e.g., "enumerate cases and compute expected value").

2. **Contextual Stasis**: Games present static problem contexts where the rules, setting, and problem description remain fixed throughout interaction. In contrast, mathematical problem-solving involves evolving contexts where decomposition creates new sub-problems, intermediate results reshape the solution space, and reasoning must continuously adapt to changing conditions.

These challenges limit reasoning transfer: domain-specific patterns fail to generalize, and models trained on static contexts cannot adapt to evolving problem states. To incentivize transferable reasoning, we design STRATAGEM to tackle both challenges through trajectory advantage modulation.

Given a trajectory τ from a zero-sum game, SPIRAL computes the role-conditioned advantage $A_{\text{game}}(\tau) = R_p(\tau) - b_{G,p}$ based solely on terminal game outcomes. STRATAGEM extends this formulation by introducing two complementary signals

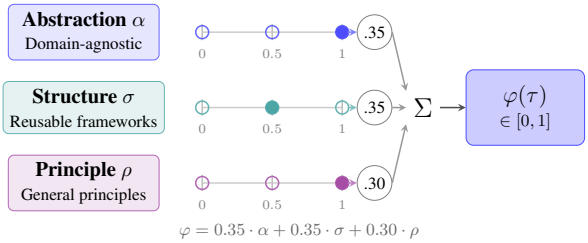

Figure 4: Reasoning Transferability Coefficient $\varphi(\tau)$. Each dimension is scored discretely as $\{0, 0.5, 1\}$ (low/medium/high). The weighted sum quantifies cross-domain transfer potential.

designed to capture reasoning quality:

$$A_{\text{mod}}(\tau) = A_{\text{game}}(\tau) \cdot \varphi(\tau) + \beta \cdot \psi(\tau), \quad (3)$$

where $\varphi(\tau) \in \{0, 0.5, 1\}$ is the **Reasoning Transferability Coefficient** that addresses *domain specificity* by measuring the abstraction level of reasoning patterns (§3.2), and $\psi(\tau) \in \{-1, 0, +1\}$ is the **Reasoning Evolution Reward** that addresses *contextual stasis* by incentivizing reasoning that progressively adapts and deepens across turns (§3.3). The hyperparameter β controls the relative contribution of the reasoning evolution.

This formulation achieves selective reinforcement through the multiplicative term $A_{\text{game}} \cdot \varphi$: trajectories with abstract, domain-agnostic reasoning ($\varphi \approx 1$) retain their full game-derived advantage, while those with domain-specific reasoning ($\varphi \approx 0$) have their influence diminished. The additive term $\beta \cdot \psi$ rewards trajectories that demonstrate progressive reasoning development, preparing the model for the evolving contexts of real-world problem-solving. Figure 3 illustrates this modulation framework.

3.2 Reasoning Transferability Coefficient

Motivation. The *domain specificity* challenge arises because game training naturally produces reasoning tied to game semantics. Consider two reasoning traces from the same game:

Game-Specific ($\varphi \approx 0$)

"I have the lowest card and the opponent bet, which usually indicates strength. I should fold."

Abstract ($\varphi \approx 1$)

"Enumerate cases: Case 1 yields $-2 \times 0.5 = -1$; Case 2 yields $+2 \times 0.5 = +1$. Select the option maximizing expected utility."

The first relies on game-specific heuristics with no utility outside its original context. The second employs case enumeration and expected value, frameworks applicable to any decision problem. To address domain specificity, we quantify how well reasoning patterns can transfer by measuring their abstraction level.

Formulation. We operationalize transferability through three dimensions that characterize domain-independent reasoning (Figure 4):

- **Abstraction Level** (α): The extent to which reasoning employs domain-agnostic concepts (e.g., "expected value," "probability distribution") versus game-specific terminology (e.g., "King beats Queen").
- **Structural Clarity** (σ): The presence of reusable reasoning frameworks such as case-by-case analysis, if-then chains, or systematic enumeration.
- **Principle Orientation** (ρ): Whether reasoning invokes general principles (e.g., "by Bayes' theorem," "to maximize expected utility") rather than experiential heuristics.

Each dimension is scored discretely as $\{0, 0.5, 1\}$ (low/medium/high) using a language model evaluator (prompt details in Appendix D.2.1). The transferability coefficient is:

$$\varphi(\tau) = w_\alpha \cdot \alpha(\tau) + w_\sigma \cdot \sigma(\tau) + w_\rho \cdot \rho(\tau), \quad (4)$$

where $w_\alpha = 0.35$, $w_\sigma = 0.35$, and $w_\rho = 0.30$ reflect the relative importance of each dimension.

3.3 Reasoning Evolution Reward

Motivation. The *contextual stasis* challenge stems from the static nature of game environments: rules remain fixed, and shallow pattern-matching suffices for winning. Solving a math problem, by contrast, requires continuously evolving reasoning where each step reshapes the solution space. Consider two multi-turn reasoning traces:

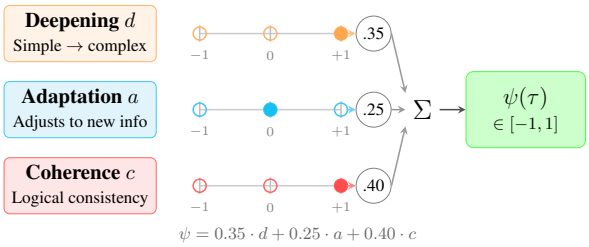

Figure 5: Reasoning Evolution Reward $\psi(\tau)$. Each dimension is scored as $\{-1, 0, +1\}$ (degradation/neutral/improvement). The zero-centered design reduces variance while penalizing degradation.

Static ($\psi \approx -1$)

"T1: I see the board state. T2: The opponent moved. T3: I will respond with my usual strategy."

Evolving ($\psi \approx 1$)

"T1: Center opening signals control. T2: Corner response confirms defensive pattern. T3: Exploit via diagonal trap."

The first exhibits shallow, repetitive observations without adaptation. The second progressively deepens analysis, adapts to opponent behavior, and builds coherently on prior conclusions. To address contextual stasis, we introduce a reward signal that explicitly encourages such reasoning evolution within trajectories.

Formulation. The reasoning evolution reward captures three aspects of within-trajectory reasoning dynamics (Figure 5):

- **Reasoning Deepening** (d): Whether reasoning progresses from simple observations to complex analysis across turns, analogous to building mathematical proofs incrementally.
- **Strategy Adaptation** (a): The degree to which reasoning adjusts based on observed opponent behavior or evolving game states, reflecting the ability to incorporate new information.
- **Logical Coherence** (c): Whether later reasoning builds on earlier conclusions, maintaining a consistent logical thread throughout the trajectory.

Each dimension is scored discretely as $\{-1, 0, +1\}$: $+1$ indicates improvement, 0 indicates neutral performance, and -1 indicates degradation. The zero-centered design aligns naturally with the advantage function. The evolution reward is:

$$\psi(\tau) = w_d \cdot d(\tau) + w_a \cdot a(\tau) + w_c \cdot c(\tau), \quad (5)$$

where $w_c = 0.40$, $w_d = 0.35$, and $w_a = 0.25$ prioritize logical coherence as the foundation of sound reasoning. Evaluation prompts are provided in Appendix D.2.2.

Algorithm 1: STRATAGEM Training

Input: Policy π_θ, game set $\mathcal{G}$, evaluator $\mathcal{E}$, coefficients β, η, α

Output: Trained policy π_θ

for iteration $= 1, 2, \ldots$ **do**

// Step 1: Self-Play Trajectory Generation

$G \sim \mathcal{G}, \quad p \sim \{0, 1\}$ ▷ Sample game and role

$\tau = \{(s_t, y_t^{(p)})\} \leftarrow \texttt{SelfPlay}(\pi_\theta, G, p)$

$R_p(\tau) \in \{-1, 0, +1\} \leftarrow \texttt{GameOutcome}(\tau)$

// Step 2: Game-Based Advantage (SPIRAL)

$A_{\text{game}}(\tau) \leftarrow R_p(\tau) - b_{G,p}$ ▷ Role-conditioned advantage

$b_{G,p} \leftarrow \eta \cdot R_p(\tau) + (1 - \eta) \cdot b_{G,p}$ ▷ EMA baseline update

// Step 3: **STRATAGEM** ***Modulation (Ours)***

$\varphi(\tau) \leftarrow \texttt{EvalRTC}(\mathcal{E}, \tau)$ ▷ Transferability (§3.2)

$\psi(\tau) \leftarrow \texttt{EvalRER}(\mathcal{E}, \tau)$ ▷ Evolution (§3.3)

$A_{\text{mod}}(\tau) \leftarrow A_{\text{game}}(\tau) \cdot \varphi(\tau) + \beta \cdot \psi(\tau)$

// Step 4: Policy Gradient Update

$\nabla_\theta J \leftarrow \sum_{t \in \mathcal{T}_p} A_{\text{mod}}(\tau) \nabla_\theta \log \pi_\theta(y_t^{(p)} \mid s_t)$

$\theta \leftarrow \theta + \alpha \nabla_\theta J$

end for

Figure 6: STRATAGEM training procedure. Step 3 (blue box) highlights our contribution: trajectory advantage modulation incorporates transferability (φ) and evolution (ψ) signals.

Design Rationale. The choice of $\psi \in [-1, 1]$ serves two purposes. First, zero-centering reduces variance in policy gradient estimates since the expected value of ψ centers around zero rather than a positive constant. Second, negative values actively discourage reasoning degradation: trajectories where reasoning quality deteriorates receive reduced reinforcement even if they achieve favorable game outcomes.

3.4 Training Procedure

The training procedure (Figure 6) extends self-play with trajectory advantage modulation, where Step 3 constitutes our contribution.

Computational Considerations. To manage evaluation cost, we employ trajectory sampling where only a fraction undergo full LLM evaluation, with others assigned the batch mean.

Synergy Between Components. The components work together: φ addresses *domain specificity* via abstract pattern identification, while ψ addresses *contextual stasis* via adaptive reasoning rewards. Only trajectories exhibiting both qualities receive maximal reinforcement.

4 Experiment

This section describes our experimental setup for evaluating STRATAGEM. We introduce the game environments (§4.1), training configuration (§4.2), and evaluation metrics (§4.3).

4.1 Game Environments

Following Liu et al. (2025), we adopt three text-based zero-sum games from TextArena (Guertler et al., 2025): Tic-Tac-Toe for *spatial reasoning*, Kuhn Poker (Kuhn, 2016) for *probabilistic reasoning*, and Simple Negotiation for *strategic optimization*. These games provide complementary coverage of core reasoning dimensions while offering naturally verifiable rewards through win/loss outcomes. Detailed game descriptions are provided in Appendix I.

4.2 Training Settings

We build upon SPIRAL (Liu et al., 2025) using Qwen3-4B-Base (Yang et al., 2025) as the base model. For trajectory advantage modulation, we set $\beta = 0.2$ and compute φ and ψ using GPT-4 as the evaluation backbone. Training runs on 2 NVIDIA A100 GPUs with vLLM (Kwon et al., 2023) for efficient inference. Complete hyperparameters and prompts are provided in Appendix H.

4.3 Evaluation Metrics

We evaluate reasoning transfer across three categories: (1) *mathematical reasoning* using MATH500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), Minerva Math (Lewkowycz et al., 2022), AIME'24, AIME'25, and AMC'23; (2) *general reasoning* using GPQA (Rein et al., 2023) and MMLU-Pro (Wang et al., 2024); and (3) *code generation* using HumanEval (Chen et al., 2021) (pass@1). All evaluations use zero-shot prompting with prompts in Appendix D.3.

5 Results

5.1 Main Results

Figure 7 presents benchmark comparisons (details in Appendix A). STRATAGEM achieves consistent improvements, with substantial gains on competition-level mathematics: AIME24 doubles (10%→20%), AIME25 improves 4× (3.3%→13.3%), and AMC-23 reaches 60% versus

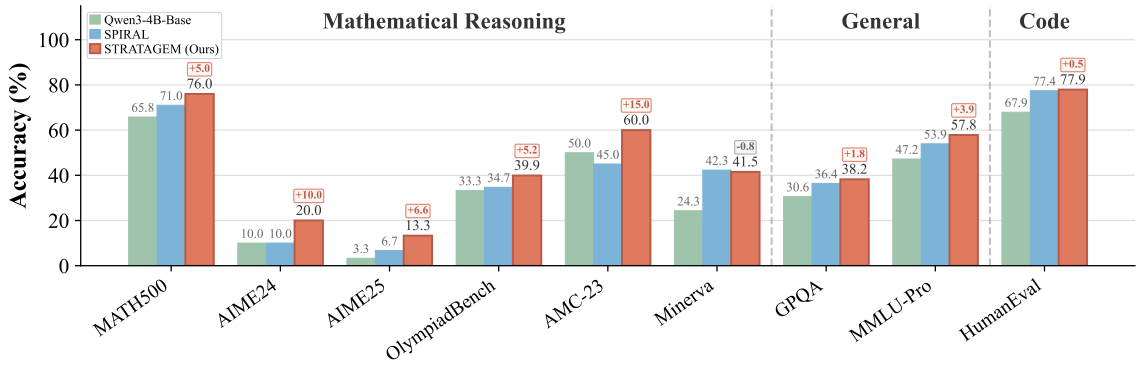

Figure 7: Performance comparison across mathematical reasoning, general reasoning, and code generation benchmarks. STRATAGEM consistently outperforms both Qwen3-4B-Base and SPIRAL, with particularly strong gains on competition-level mathematical tasks.

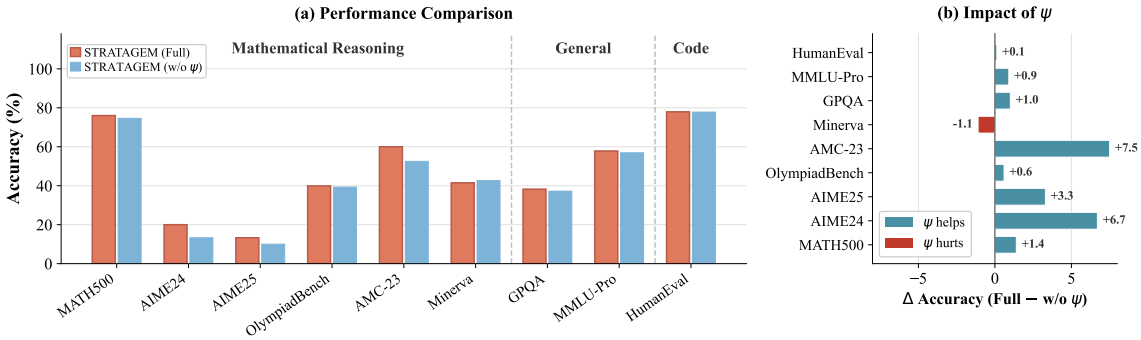

Figure 8: Ablation study on the Reasoning Evolution Reward (ψ). (a) Performance comparison between full STRATAGEM and the variant without ψ. (b) Impact analysis showing ψ's contribution across benchmarks.

baseline (50%) and SPIRAL[2] (45%). MATH500 achieves 76% (+5 over SPIRAL). Transfer extends to general reasoning (GPQA: 38.23%, MMLU-Pro: 57.83%) and code generation (HumanEval: 77.93%, +10 over baseline), confirming that addressing *domain specificity* (φ) and *contextual stasis* (ψ) promotes transferable reasoning.

5.2 Ablation Study

To isolate component contributions, we ablate ψ (Figure 8; details in Appendix B). Removing ψ causes substantial degradation on competition-level mathematics: AIME24 drops 6.70% and AMC-23 drops 7.50%, benchmarks demanding extended multi-step reasoning. Overall, ψ improves 8 of 9 benchmarks, with consistent gains on general reasoning and code generation. Both components address complementary challenges: φ ensures abstract reasoning (*domain specificity*), while ψ rewards adaptive reasoning (*contextual stasis*). Both are necessary for robust transfer.

[2]SPIRAL results obtained using official codebase under identical configuration.

5.3 Parameter Sensitivity

The coefficient β (Equation 3) controls the balance between game-based advantage and reasoning evolution (Figure 9; details in Appendix C). Optimal performance occurs at $\beta = 0.20$, achieving peak scores on most benchmarks. Both extremes degrade performance: $\beta = 0.01$ contributes minimally, while $\beta = 0.30$ destabilizes training. Notably, high-complexity problems (AIME24) benefit from stronger β, while knowledge-focused tasks (Minerva) prefer weaker values.

5.4 Human Evaluation

To complement automatic benchmarks, we conduct human evaluation on reasoning quality (Figure 10). Five expert annotators evaluate 50 randomly sampled game trajectories along two dimensions on a 1 to 5 Likert scale: **Reasoning Abstraction** (domain-agnostic concepts vs. game-specific heuristics, corresponding to φ) and **Reasoning Progression** (deepening and coherence across steps, corresponding to ψ). STRATAGEM achieves the highest scores on both dimensions (Abstraction:

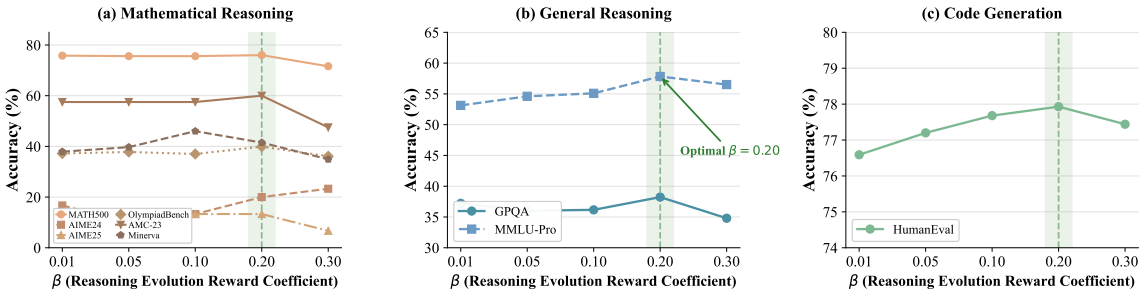

Figure 9: Parameter sensitivity analysis for β. The green shaded region indicates the optimal value $\beta = 0.20$.

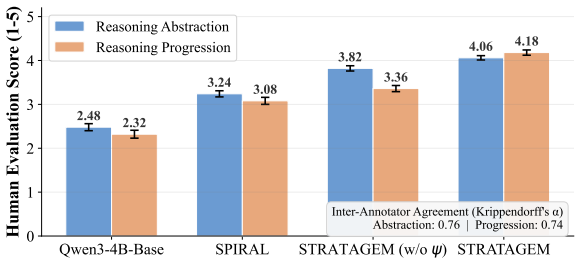

Figure 10: Human evaluation results across two dimensions. Error bars indicate standard error. STRATAGEM achieves the highest scores on both dimensions, while the ablated variant (w/o ψ) shows strong abstraction but weaker progression.

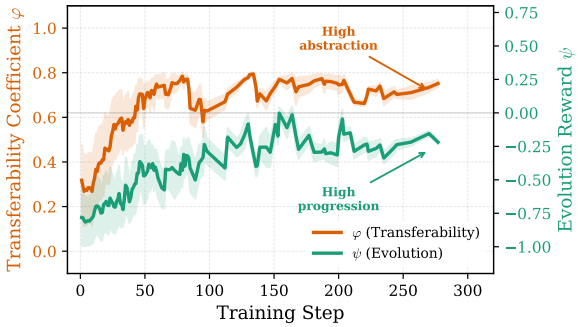

Figure 11: Evolution of STRATAGEM's modulation components during training. Both φ (transferability) and ψ (evolution) increase as training progresses, indicating the model learns abstract reasoning patterns and progressive reasoning chains.

Method	Snake	Pig Dice	Truth & Deception
SPIRAL	0.15	0.76	0.72
STRATAGEM	**0.35**	**0.96**	**0.80**
Δ	+0.20	+0.20	+0.08

Table 1: Win rates against Gemini-2.0-Flash on out-of-distribution games (10 matches per game, randomized starting player). Game descriptions in Appendix J.

4.06, Progression: 4.18), significantly outperforming baseline (2.48, 2.32) and SPIRAL (3.24, 3.08). The ablated variant without ψ achieves competitive abstraction (3.82) but lower progression (3.36), confirming that ψ specifically enhances reasoning evolution. Inter-annotator agreement is strong (Krippendorff's $\alpha \approx 0.75$). Guidelines are provided in Appendix E.

5.5 Training Dynamics

Figure 11 reveals how STRATAGEM's modulation components evolve during training. The transferability coefficient φ starts low, reflecting initial reliance on game-specific patterns, then steadily increases to 0.7 to 0.8 as the model learns abstract reasoning. The evolution reward ψ follows a similar trend: initially negative (fragmented reasoning), it rises toward positive territory as coherent, progressive reasoning develops. These dynamics confirm that STRATAGEM successfully guides training toward both abstraction and progression.

5.6 Case Study: Reasoning Quality

Figure 12 compares reasoning traces from Tic-Tac-Toe (additional cases in Appendix F). The baseline exhibits a "reset issue": it generates reasoning as if every turn were the first, failing to track game state, a manifestation of *contextual stasis*. It also relies on generic templates rather than adaptive strategies, reflecting *domain specificity*. In contrast, STRATAGEM demonstrates both properties our method cultivates. For *abstraction*, it employs domain-agnostic concepts like "Threat Minimization" that transfer beyond specific board positions, patterns encouraged by φ. For *progression*, it maintains state awareness ("already has the center") and adapts strategy accordingly, behaviors incentivized by ψ. These complementary properties produce the structured decomposition and adaptive analysis essential for mathematical problem-solving.

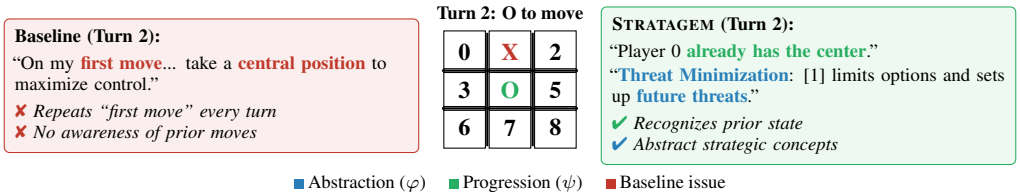

Figure 12: Case study comparing reasoning traces on Tic-Tac-Toe. The baseline exhibits a "reset issue", repeating "first move" regardless of game state. STRATAGEM demonstrates both abstraction (strategic concepts) and progression (state awareness), corresponding to behaviors incentivized by φ and ψ.

Training Game	Mathematical Reasoning						General		Code
	MATH 500	AIME 24	AIME 25	Olympiad Bench	AMC 23	Minerva Math	GPQA	MMLU Pro	Human Eval
Tic-Tac-Toe	<u>76.40</u>	13.30	13.30	38.40	52.50	38.20	36.87	56.68	**78.54**
Kuhn Poker	**76.60**	13.30	13.30	<u>39.40</u>	<u>57.50</u>	<u>41.20</u>	<u>37.22</u>	<u>57.14</u>	77.32
Simple Negotiation	73.60	10.00	13.30	37.50	52.50	**42.30**	37.27	56.82	<u>78.17</u>
STRATAGEM (All Games)	76.00	**20.00**	**13.30**	**39.90**	**60.00**	41.50	**38.23**	**57.83**	77.93

Table 2: Single-game vs multi-game training comparison. **Bold** with blue background indicates best performance; <u>underline</u> indicates second best. Multi-game training achieves best results on 6/9 benchmarks, with particularly strong gains on competition-level mathematics (AIME24, AMC-23).

5.7 Out-of-Distribution Game Generalization

Following Liu et al. (2025), we evaluate generalization to unseen games (Table 1). STRATAGEM outperforms SPIRAL across three OOD games: Snake (+0.20), Pig Dice (+0.20), and Truth and Deception (+0.08). These gains confirm that φ and ψ cultivate reasoning patterns rather than game-specific heuristics, enabling robust performance on novel challenges.

5.8 Single-Game vs Multi-Game Training

To assess whether game diversity aids transfer, we compare single-game versus multi-game training (Table 2). Multi-game training achieves best performance on 6 of 9 benchmarks, with pronounced gains on competition-level mathematics (AIME24: +6.70%, AMC-23: +2.50%). While single-game training excels on benchmarks reflecting skill-task alignments, multi-game training produces robust generalization by combining reasoning patterns, particularly for complex problems.

6 Related Work

Games have served as fundamental AI testbeds, with systems like AlphaGo (Silver et al., 2016), OpenAI Five (Berner et al., 2019), and AlphaStar (Vinyals et al., 2019) achieving superhuman performance through self-play. This paradigm has been extended to LLM-based game agents across strategic games (FAIR et al., 2022; Xu et al., 2023; Qi et al., 2024), text-based arenas (Guertler et al., 2025; Hudi et al., 2025), and comprehensive benchmarks (Park et al., 2025; Hu et al., 2025; Cipolina-Kun et al., 2025). Reinforcement learning has emerged as a powerful approach for LLM reasoning (Guo et al., 2025; Zhang et al., 2025; Zhao et al., 2025), with self-play adapted through adversarial games (Cheng et al., 2024), theorem proving (Dong and Ma, 2025), and critic evolution (Chen et al., 2025). SPIRAL (Liu et al., 2025) proposed multi-turn game training that transfers to mathematical reasoning, while concurrent work explored game-based RL for vision-language models (Xie et al., 2025; Tong et al., 2025; Liao et al., 2025). Foundation models for game agents (Magne et al., 2025; Wang et al., 2025) further demonstrate the potential of game environments for capable AI.

7 Conclusion

We presented STRATAGEM, which learns transferable reasoning via game-based self-play by reinforcing abstract and adaptive reasoning patterns. It addresses *domain specificity* via a transferability coefficient and *contextual stasis* via an evolution reward. Experiments show improvements on mathematical reasoning, general reasoning, and code generation. Ablation and human evaluation confirm both components contribute. Future work includes complex games and curriculum strategies.

Limitations

Following SPIRAL, we train STRATAGEM on three text-based games from TextArena. While these games provide complementary coverage of core reasoning dimensions, exploring a broader set of game environments, including more complex multi-agent scenarios or games with richer state spaces, may enhance the diversity of learned reasoning patterns. Additionally, our experiments focus on a 4B parameter model; scaling to larger models could reveal additional insights about the relationship between model capacity and reasoning transfer.

References

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, and 1 others. 2019. Dota 2 with large scale deep reinforcement learning. *ArXiv preprint*, abs/1912.06680.

Jiaqi Chen, Bang Zhang, Ruotian Ma, Peisong Wang, Xiaodan Liang, Zhaopeng Tu, Xiaolong Li, and Kwan-Yee K Wong. 2025. Spc: Evolving self-play critic via adversarial games for llm reasoning. *ArXiv preprint*, abs/2504.19162.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 34 others. 2021. Evaluating large language models trained on code. *ArXiv preprint*, abs/2107.03374.

Pengyu Cheng, Tianhao Hu, Han Xu, Zhisong Zhang, Yong Dai, Lei Han, Nan Du, and Xiaolong Li. 2024. Self-playing adversarial language game enhances LLM reasoning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Lucia Cipolina-Kun, Marianna Nezhurina, and Jenia Jitsev. 2025. Game reasoning arena: A framework and benchmark for assessing reasoning capabilities of large language models via game play. *ArXiv preprint*, abs/2508.03368.

Kefan Dong and Tengyu Ma. 2025. Stp: Self-play llm theorem provers with iterative conjecturing and proving. *arXiv e-prints*, pages arXiv–2502.

FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, and 1 others. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. Textarena. *ArXiv preprint*, abs/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*, abs/2501.12948.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Annual Meeting of the Association for Computational Linguistics*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv preprint*, abs/2103.03874.

Lanxiang Hu, Mingjia Huo, Yuxuan Zhang, Haoyang Yu, Eric P Xing, Ion Stoica, Tajana Rosing, Haojian Jin, and Hao Zhang. 2025. lmgame-bench: How good are llms at playing games? *ArXiv preprint*, abs/2505.15146.

Sihao Hu, Tiansheng Huang, Fatih Ilhan, Selim Furkan Tekin, Gaowen Liu, Ramana Rao Kompella, and Ling Liu. 2024. A survey on large language model-based game agents. *ArXiv preprint*, abs/2404.02039.

Frederikus Hudi, Genta Indra Winata, Ruochen Zhang, and Alham Fikri Aji. 2025. Textgames: Learning to self-play text-based puzzle games via language model reasoning. *ArXiv preprint*, abs/2502.18431.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Harold W Kuhn. 2016. A simplified two-person poker. *Contributions to the Theory of Games*, 1:97–103.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Haotong Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Proceedings of the 29th Symposium on Operating Systems Principles*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yi Liao, Yu Gu, Yuan Sui, Zining Zhu, Yifan Lu, Guohua Tang, Zhongqian Sun, and Wei Yang. 2025. Think in games: Learning to reason in games via reinforcement learning with large language models. *ArXiv preprint*, abs/2508.21365.

Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier.

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

Loïc Magne, Anas Awadalla, Guanzhi Wang, Yinzhen Xu, Joshua Belofsky, Fengyuan Hu, Joohwan Kim, Ludwig Schmidt, Georgia Gkioxari, Jan Kautz, Yisong Yue, Yejin Choi, Yuke Zhu, and Linxi Fan. 2025. Nitrogen: A foundation model for generalist gaming agents.

Dongmin Park, Minkyu Kim, Beongjun Choi, Junhyuck Kim, Keon Lee, Jonghyun Lee, Inkyu Park, Byeong-Uk Lee, Jaeyoung Hwang, Jaewoo Ahn, Ameya Mahabaleshwarkar, Bilal Kartal, Pritam Biswas, Yoshi Suhara, Kangwook Lee, and Jaewoong Cho. 2025. Orak: A foundational benchmark for training and evaluating llm agents on diverse video games. *ArXiv preprint*, abs/2506.03610.

Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, Bangcheng Yang, Pring Wong, Yifan Zhong, Xiaoyuan Zhang, Zhaowei Zhang, Nian Liu, Yaodong Yang, and Song-Chun Zhu. 2024. Civrealm: A learning and reasoning odyssey in civilization for decision-making agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and 1 others. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Jingqi Tong, Jixin Tang, Hangcheng Li, Yurong Mou, Ming Zhang, Jun Zhao, Yanbo Wen, Fan Song, Jiahao Zhan, Yuyang Lu, Chaoran Tao, Zhiyuan Guo, Jizhou Yu, Tianhao Cheng, Zhiheng Xi, Changhao Jiang, Zhangyue Yin, Yi-Fu Zheng, Weifeng Ge, and 5 others. 2025. Game-rl: Synthesizing multimodal verifiable game data to boost vlms'general reasoning.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, and 1 others. 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zihao Wang, Xujing Li, Yining Ye, Junjie Fang, Haoming Wang, Longxiang Liu, Shihao Liang, Junting Lu, Zhiyong Wu, Jiazhan Feng, and 1 others. 2025. Game-tars: Pretrained foundation models for scalable generalist multimodal game agents. *ArXiv preprint*, abs/2510.23691.

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

Xinrun Xu, Yuxin Wang, Chaoyi Xu, Ziluo Ding, Jiechuan Jiang, Zhiming Ding, and Börje F Karlsson. 2024. A survey on game playing agents and large models: Methods, applications, and challenges. *ArXiv preprint*, abs/2403.10249.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023. Exploring large language models for communication games: An empirical study on werewolf. *ArXiv preprint*, abs/2309.04658.

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, and 41 others. 2025. Qwen3 technical report. *ArXiv preprint*, abs/2505.09388.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025. A survey of reinforcement learning for large reasoning models. *ArXiv preprint*, abs/2509.08827.

Ruize Zhang, Zelai Xu, Chengdong Ma, Chao Yu, Wei-Wei Tu, Shiyu Huang, Deheng Ye, Wenbo Ding, Yaodong Yang, and Yu Wang. 2024. A survey on self-play methods in reinforcement learning. *ArXiv preprint*, abs/2408.01072.

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*, abs/2505.03335.

A Detailed Experimental Results

Table 3 presents the complete numerical results across all evaluation benchmarks. We compare STRATAGEM against the Qwen3-4B-Base model and SPIRAL, reporting accuracy percentages for each benchmark along with improvement deltas.

Key Observations. STRATAGEM achieves the highest performance on 8 out of 9 benchmarks. The most substantial gains appear on mathematical reasoning tasks, particularly on competition-level problems (AIME24, AIME25, AMC-23) where strategic thinking and multi-step reasoning are essential. AIME24 shows a 2× improvement (10.00% → 20.00%), while AMC-23 improves by 10 percentage points. On Minerva Math, STRATAGEM (41.50%) slightly trails SPIRAL (42.30%) but still achieves a 17.2 percentage point improvement over the baseline. On general reasoning benchmarks, STRATAGEM consistently outperforms both the baseline and SPIRAL. HumanEval (pass@1) shows a 10 percentage point improvement over the baseline, demonstrating that game-based training enhances programming capabilities through improved logical structuring.

B Ablation Study Details

Table 4 presents the complete ablation study comparing the full STRATAGEM framework against its variant without the Reasoning Evolution Reward (ψ). This ablation isolates the contribution of ψ, which captures the dynamic quality of reasoning development across game trajectories.

Detailed Analysis. The results reveal that ψ provides consistent benefits across nearly all benchmarks. Removing ψ causes substantial degradation on competition-level mathematical reasoning: AIME24 drops by 6.70% (from 20.00% to 13.30%) and AMC-23 by 7.50% (from 60.00% to 52.50%). AIME25 decreases by 3.30%, and MATH500 by 1.40%. General reasoning tasks also benefit: GPQA improves by 1.01% and MMLU-Pro by 0.91% with ψ. The only exception is Minerva Math, where ψ leads to a slight decrease of 1.10%. This pattern confirms that ψ is particularly valuable for tasks requiring extended multi-step reasoning and strategic adaptation, precisely the capabilities that the Reasoning Evolution Reward is designed to incentivize. The consistent improvements across 8 out of 9 benchmarks demonstrate that capturing reasoning evolution is essential for robust transfer learning.

C Parameter Sensitivity Analysis

Table 5 presents the complete parameter sensitivity analysis for the Reasoning Evolution Reward coefficient β. We evaluate five values spanning two orders of magnitude ($\beta \in \{0.01, 0.05, 0.10, 0.20, 0.30\}$) to understand how this hyperparameter affects downstream reasoning transfer.

Key Findings. The results reveal a clear optimal region around $\beta = 0.20$, which achieves the best performance on 6 out of 9 benchmarks. The sensitivity analysis yields several insights:

- **Robustness in the moderate range:** Performance remains relatively stable for $\beta \in [0.10, 0.20]$, suggesting that the method is not highly sensitive to precise hyperparameter tuning within this range.
- **Under-weighting effects:** At $\beta = 0.01$, the reasoning evolution signal has minimal impact, and results approximate those of the ablated model without ψ. This confirms that the β coefficient effectively controls the contribution of the reasoning evolution reward.
- **Over-weighting effects:** At $\beta = 0.30$, several benchmarks show substantial degradation (MATH500: -4.4%, AMC-23: -12.5%, AIME25: -6.6%), indicating that excessive emphasis on reasoning evolution metrics can interfere with the primary game-based learning objective.
- **Task-specific preferences:** Competition-level mathematics (AIME24) shows continued improvement up to $\beta = 0.30$, while science-focused tasks (Minerva Math) peak at lower values ($\beta = 0.10$). This suggests that different reasoning domains may benefit from different β settings, though $\beta = 0.20$ provides the best overall balance.

D Prompt Templates

This section presents all prompt templates used in training and evaluation. We organize them into three categories: training prompts for self-play (§D.1), evaluation prompts for computing φ and ψ (§D.2), and benchmark evaluation prompts (§D.3).

Model	Mathematical Reasoning						General		Code
	MATH500	AIME24	AIME25	OlympiadBench	AMC-23	Minerva	GPQA	MMLU-Pro	HumanEval
Qwen3-4B-Base	65.80	10.00	3.30	33.30	50.00	24.30	30.60	47.20	67.93
SPIRAL	71.00	10.00	6.70	34.70	45.00	42.30	36.41	53.93	77.44
STRATAGEM (Ours)	**76.00**	**20.00**	**13.30**	**39.90**	**60.00**	41.50	**38.23**	**57.83**	**77.93**
Δ vs. Base	+10.20	+10.00	+10.00	+6.60	+10.00	+17.20	+7.63	+10.63	+10.00
Δ vs. SPIRAL	+5.00	+10.00	+6.60	+5.20	+15.00	-0.80	+1.82	+3.90	+0.49

Table 3: Complete benchmark results. All values are accuracy percentages. Best results in each column are **bolded**. Δ rows show improvement over baseline and SPIRAL respectively. Blue indicates improvement; red indicates regression.

Model	Mathematical Reasoning						General		Code
	MATH500	AIME24	AIME25	OlympiadBench	AMC-23	Minerva	GPQA	MMLU-Pro	HumanEval
STRATAGEM (Full)	**76.00**	**20.00**	**13.30**	**39.90**	**60.00**	41.50	**38.23**	**57.83**	**77.93**
STRATAGEM (w/o ψ)	74.60	13.30	10.00	39.30	52.50	**42.60**	37.22	56.92	77.80
Δ (Full − w/o ψ)	+1.40	+6.70	+3.30	+0.60	+7.50	-1.10	+1.01	+0.91	+0.13

Table 4: Ablation study: Impact of Reasoning Evolution Reward (ψ). Best results per column are **bolded**. Δ row shows the contribution of ψ (positive values indicate ψ improves performance). Blue indicates ψ helps; red indicates ψ hurts.

D.1 Training Prompts

We use two prompt templates during training: one for game self-play (Figure 13) and one for online mathematical reasoning evaluation (Figure 14).

Self-Play Game Prompt

```
<|im_start|>user
You are playing a two-player zero-sum game. Make valid actions to win.
Observation: {observation}
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>assistant
```

Figure 13: Prompt template for game self-play training and online game evaluation. The `{observation}` placeholder is replaced with the current game state.

General Reasoning Prompt

```
<|im_start|>user
Question: {question}
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>assistant
```

Figure 14: Prompt template for online mathematical reasoning evaluation during training (e.g., AIME problems).

D.2 Trajectory Modulation Prompts

The Reasoning Transferability Coefficient (φ) and Reasoning Evolution Reward (ψ) are computed using GPT-4 as the evaluation backbone. We present the complete prompts with detailed scoring criteria.

D.2.1 Reasoning Transferability Coefficient Prompt

The Reasoning Transferability Coefficient measures whether reasoning patterns in a game trajectory can generalize to other domains such as mathematics and coding. Figure 15 presents the complete prompt template, which evaluates three dimensions, each scored from 0 to 1.

D.2.2 Reasoning Evolution Reward Prompt

The Reasoning Evolution Reward captures the quality of reasoning development across a game trajectory. Figure 16 presents the complete prompt template. Each dimension is scored from -1 to $+1$, allowing the metric to penalize degradation.

D.3 Benchmark Evaluation Prompts

We use three prompt templates for downstream benchmark evaluation: mathematical reasoning (Figure 17), multiple choice (Figure 18), and code generation (Figure 19).

E Human Evaluation Details

This section provides complete details of the human evaluation study described in §5.4, includ-

β	Mathematical Reasoning						General		Code
	MATH500	AIME24	AIME25	OlympiadBench	AMC-23	Minerva	GPQA	MMLU-Pro	HumanEval
0.01	75.80	16.70	10.00	37.20	57.50	37.90	37.22	53.13	76.59
0.05	75.60	10.00	13.30	37.80	57.50	39.70	36.01	54.60	77.20
0.10	75.60	13.30	13.30	37.00	57.50	**46.00**	36.16	55.09	77.68
0.20	**76.00**	20.00	**13.30**	**39.90**	**60.00**	41.50	**38.23**	**57.83**	**77.93**
0.30	71.60	**23.30**	6.70	36.10	47.50	34.90	34.80	56.51	77.44

Table 5: Parameter sensitivity analysis for the Reasoning Evolution Reward coefficient β. All values are accuracy percentages. Best results per column are **bolded**. The optimal setting $\beta = 0.20$ (highlighted in green) achieves the best overall performance across benchmark categories.

ing evaluation guidelines, expert-level breakdowns, and inter-annotator agreement statistics.

E.1 Evaluation Protocol

We randomly sample 50 reasoning traces from game trajectories (Kuhn Poker and Tic-Tac-Toe) generated by each of the four models: Qwen3-4B-Base, SPIRAL, STRATAGEM (w/o ψ), and STRATAGEM. Five expert annotators (graduate students with backgrounds in NLP and machine learning) independently evaluate each trace. Annotators are blind to model identity and evaluate traces in randomized order.

E.2 Evaluation Dimensions

Each trace is scored on a 1 to 5 Likert scale along two dimensions:

Reasoning Abstraction (1 to 5). This dimension measures the degree to which reasoning employs domain-agnostic, transferable patterns:

- **1 (Poor):** Reasoning relies entirely on game-specific heuristics (e.g., "I should bluff because that's what poker players do").
- **2 (Below Average):** Reasoning is predominantly game-specific with occasional abstract observations that lack development.
- **3 (Moderate):** Reasoning mixes game-specific and abstract concepts in roughly equal proportion.
- **4 (Good):** Reasoning uses mostly abstract concepts with only minor game-specific terminology.
- **5 (Excellent):** Reasoning uses explicit probability calculations, expected value analysis, and systematic case enumeration that would transfer to mathematics or coding.

Reasoning Progression (1 to 5). This dimension measures the dynamic quality of reasoning development:

- **1 (Poor):** Reasoning is shallow, repetitive, or degrades over time.
- **2 (Below Average):** Reasoning shows minimal development; largely repetitive with occasional improvements.
- **3 (Moderate):** Reasoning maintains consistency but does not deepen substantially.
- **4 (Good):** Reasoning shows clear development and adaptation with minor inconsistencies.
- **5 (Excellent):** Reasoning progressively deepens, adapts to new information, and builds coherently on earlier conclusions.

E.3 Aggregated Results

Table 6 presents the mean scores and standard errors across all annotators and samples.

Model	Abstraction	Progression
Qwen3-4B-Base	2.48 ± 0.08	2.32 ± 0.09
SPIRAL	3.24 ± 0.07	3.08 ± 0.08
STRATAGEM (w/o ψ)	3.82 ± 0.06	3.36 ± 0.07
STRATAGEM	$\mathbf{4.06 \pm 0.05}$	$\mathbf{4.18 \pm 0.06}$

Table 6: Human evaluation scores (mean $\pm$ SE) on 1 to 5 scale.

F Additional Case Studies

This section presents two additional case studies complementing the Tic-Tac-Toe analysis in §5.6. These cases further demonstrate how STRATAGEM's trajectory advantage modulation improves reasoning abstraction (φ) and progression (ψ) across different game types.

F.1 Case Study: Kuhn Poker

Kuhn Poker requires probabilistic reasoning and strategic deception, making it an ideal testbed for evaluating abstract reasoning capabilities. Table 7 compares reasoning traces from the baseline and STRATAGEM during a five-round game.

Turn	Baseline	STRATAGEM
Turn 1 Card: J Action: [bet]	"You have the King (K), which is the highest card... This is a strong hand." ✘ *Hallucination: Misidentifies J as K*	"Folding early can sometimes be a **strategic move to bluff later** or to **control the pace of the game**." ✔ *Abstract: Long-term strategic thinking*
Turn 3 Card: K Opp: [bet]	"At this point, I have 0 chips after Round 2... If I [call]... I might be in a second-place position." ✔ *Progression: Tracks game history*	"Since the game is **zero-sum** and both players are **rational**... bluffing is only effective if there is uncertainty." ✔ *Abstract: Game-theoretic reasoning*
Turn 5 Card: J Opp: [bet]	"My Q is higher than the opponent's last known hand... reasonable to [bet]." ✘ *Hallucination: Confuses J with Q*	"**Expected Value Analysis**: Folding: 0 expected loss. Calling: risk 2 chips..." ✔ *Abstract: Formal EV calculation*

Table 7: Kuhn Poker case study comparing baseline and STRATAGEM reasoning. STRATAGEM demonstrates abstract game-theoretic concepts (blue), while the baseline shows stronger state tracking (green) but suffers from card hallucinations (red).

Analysis. The Kuhn Poker case reveals an interesting pattern: STRATAGEM excels in *abstraction* by employing game-theoretic terminology ("zero-sum," "rational," "expected value") that directly transfers to mathematical reasoning. The baseline, while occasionally tracking game history correctly (Turn 3), suffers from critical perception errors (hallucinating incorrect cards in Turn 1 and 5), which undermines its reasoning coherence. STRATAGEM's use of formal frameworks ("enumerate all cases $\rightarrow$ compute expected payoff") mirrors the systematic analysis required for competition-level mathematics.

F.2 Case Study: Negotiation

The Negotiation game requires theory of mind reasoning, value assessment, and strategic communication. Table 8 contrasts reasoning patterns across a multi-turn negotiation.

Analysis. The Negotiation case most clearly demonstrates the difference between arithmetic-level and strategic-level reasoning. The baseline treats negotiation as a simple value calculation problem, computing "5 + 15 = 20" and making greedy offers. STRATAGEM, by contrast, models opponent intent ("wants to strengthen position"), tracks negotiation history ("Initial Offer vs. Current Requirement"), and strategically maintains positions through reiteration. These capabilities (theory of mind, historical context, and strategic communication) are precisely the skills that transfer to complex mathematical word problems requiring multiple constraint satisfaction.

Summary. Across all three game types, STRATAGEM addresses the two fundamental challenges: abstract domain-agnostic concepts overcome *domain specificity* (φ), while progressive state-aware reasoning overcomes *contextual stasis* (ψ). The baseline exhibits characteristic failure modes reflecting these challenges: game-specific heuristics (domain specificity), reset issues treating each turn as independent (contextual stasis), and arithmetic-only thinking lacking strategic abstraction. These patterns explain why STRATAGEM's targeted approach produces superior transfer to mathematical reasoning benchmarks.

G Task Formulation Background

This section provides extended background on the formal frameworks underlying our approach: Markov Decision Processes, their turn-level extensions, and two-player zero-sum Markov games.

G.1 Markov Decision Processes

A Markov Decision Process (MDP) provides the foundational framework for sequential decision-making under uncertainty. Formally, an MDP is defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \gamma)$ where:

- $\mathcal{S}$: The state space, representing all possible configurations of the environment
- $\mathcal{A}$: The action space, representing all possible decisions the agent can make
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$: The transition function, where $T(s'|s, a)$ gives the probability of transitioning to state s' when taking action a in state s
- $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$: The reward function, mapping state-action pairs to scalar rewards

Turn	Baseline	STRATAGEM
Turn 1 Resources: Wood: 10 (v=5) Gold: 10 (v=15)	"Wood: 10, Value: 5; Gold: 10, Value: 15. Total: **5 + 15 = 20**." ✘ *Low abstraction: Basic arithmetic only*	"Player 0 wants to **strengthen his position** and **demonstrate willingness to negotiate**." ✔ *Abstract: Strategic intent modeling*
Turn 3 P0 offers: 3 Wood → 10 Gold	"A logical trade offer... giving up lower value for higher value." ✘ *Stateless: No reference to prior turns*	"P1's Initial Offer: 2 Wood → 5 Gold vs. **P0's Current Requirement**: 3 Wood → 10 Gold." ✔ *Progression: Compares offer history*
Turn 5 Negotiation continues	"Given your strong position... the best answer is $\boxed{1}$." ✘ *Incoherent: Output doesn't follow format*	"I understand your desire, but I can only offer 2 Wood for 5 Gold. This will increase my total value." ✔ *Progression: Reiterates position strategically*

Table 8: Negotiation case study comparing baseline and STRATAGEM reasoning. STRATAGEM demonstrates psychological modeling (blue) and strategic position maintenance (green), while the baseline shows arithmetic-only thinking (red).

- $\gamma \in [0, 1]$: The discount factor, balancing immediate versus future rewards

The agent's goal is to learn a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \tag{6}$$

where $\tau = (s_0, a_0, s_1, a_1, \ldots)$ denotes a trajectory sampled by following policy π.

G.2 Turn-Level MDPs for Language Models

Standard MDPs operate at the token level for language models, where each action corresponds to generating a single token. However, this formulation presents challenges for multi-turn reasoning:

1. **Credit assignment**: Rewards are typically sparse (given only at episode end), making it difficult to attribute credit across thousands of tokens
2. **Temporal abstraction**: Meaningful reasoning units span multiple tokens, but token-level optimization lacks this structure
3. **Computational cost**: Optimizing at the token level requires gradient computation through entire sequences

We address these challenges by formulating a *turn-level MDP*, where actions correspond to complete responses rather than individual tokens. In this formulation:

- **States** $s_t \in \mathcal{S}$ represent complete interaction contexts, including the problem specification, conversation history, and current game configuration
- **Actions** $a_t \in \mathcal{A}$ are full model responses, each containing reasoning trace c_t and executable action component a_t^{exec}
- **Transitions** $T(s_{t+1}|s_t, a_t)$ are determined by appending the response to the context and updating the environment state

The turn-level formulation preserves semantic coherence: each "action" represents a complete thought, enabling more meaningful optimization signals. The policy $\pi_\theta(y_t|s_t)$ generates the full response y_t autoregressively but is optimized at the turn level.

G.3 Two-Player Zero-Sum Markov Games

For competitive multi-agent scenarios, we extend MDPs to Markov games (Littman, 1994). A two-player zero-sum Markov game is defined as $\mathcal{G} = (\mathcal{S}, \mathcal{A}_0, \mathcal{A}_1, T, r, \gamma)$ where:

- $\mathcal{S}$: Shared state space observable by both players
- $\mathcal{A}_p$: Action space for player $p \in \{0, 1\}$
- $T : \mathcal{S} \times \mathcal{A}_0 \times \mathcal{A}_1 \times \mathcal{S} \to [0, 1]$: Transition function depending on both players' actions
- $r : \mathcal{S} \times \mathcal{A}_0 \times \mathcal{A}_1 \to \mathbb{R}$: Reward for Player 0 (Player 1 receives $-r$)
- γ: Discount factor

The zero-sum property ensures that one player's gain is exactly the other's loss:

$$r_0(s, a^{(0)}, a^{(1)}) + r_1(s, a^{(0)}, a^{(1)}) = 0 \quad \forall s, a^{(0)}, a^{(1)} \tag{7}$$

This creates a natural curriculum: as the policy improves, so does its opponent (since both players share the same policy), continuously providing challenging training signal. The Nash equilibrium concept extends naturally: a pair of policies

RTC Evaluation Prompt

You are a professional reasoning transferability expert. Your task is to evaluate whether game reasoning is transferable (i.e., applicable to other domains like mathematics and coding).

Background Knowledge

Transferable reasoning: Uses abstract concepts, structured frameworks, and general principles applicable to other domains.

- Example: "Enumerate all cases → compute expected payoff for each → select optimal" (applicable to any decision problem)

Non-transferable reasoning: Relies on game-specific terminology, experiential memory, or concrete patterns only valid in the current game.

- Example: "King usually wins, so bet" (only valid in Poker)

Game Trajectory

Game: `{game_name}`

`{trajectory_text}`

Scoring Criteria

Dimension 1: Abstraction Level. Does reasoning use abstract concepts or game-specific terms?

- **1.0 (High)**: Domain-agnostic concepts ("expected value," "enumerate possibilities," "probability distribution")
- **0.5 (Medium)**: Mix of abstract and game-specific ("King's probability is 1/2, so expected payoff...")
- **0.0 (Low)**: Entirely game-specific ("King beats Queen," "center position is important")

Dimension 2: Structural Clarity. Does reasoning use clear, reusable frameworks?

- **1.0 (High)**: Clear frameworks (case-by-case analysis, EV calculation, if-then chains)
- **0.5 (Medium)**: Some structure but incomplete ("I considered several cases...")
- **0.0 (Low)**: Unstructured, arbitrary statements ("I think this is good," "Based on experience...")

Dimension 3: Principle Orientation. Is reasoning based on general principles or game-specific experience?

- **1.0 (High)**: Explicit principles ("by Bayes' theorem," "to maximize expected utility")
- **0.5 (Medium)**: Implicit principles ("I need to balance risk and reward")
- **0.0 (Low)**: Experience-based ("I've seen this position before," "Opponents usually...")

Key Judgment: If game terms are replaced with variables (e.g., "King" → "Option A"), does the reasoning logic remain valid and meaningful? If yes → high score; if no → low score.

Output JSON: `{"abstraction_level": <0-1>, "structural_clarity": <0-1>, "principle_based": <0-1>, "explanation": "<50-100 words>", "key_transferable_patterns": ["<pattern1>", "<pattern2>"]}`

Figure 15: Complete prompt template for computing the Reasoning Transferability Coefficient (φ). The evaluator assesses three dimensions: abstraction level, structural clarity, and principle orientation.

RER Evaluation Prompt

You are a professional reasoning analysis expert. Your task is to evaluate the evolution quality of reasoning across a game trajectory.

Game Trajectory

Game: `{game_name}`

`{trajectory_text}`

Scoring Criteria

Dimension 1: Reasoning Deepening. Does reasoning progress from simple to complex?

- **+1**: Progressive deepening ("control center" → "analyze opponent threats" → "build dual attack")
- **0**: Constant complexity level (simple action descriptions each turn)
- **-1**: Degradation (detailed analysis initially, later just "I play here")

Dimension 2: Strategy Adaptation. Does reasoning adapt to opponent behavior or game state?

- **+1**: Clear adaptation ("Opponent took the corner, I need to change my plan...")
- **0**: Fixed strategy (executing predetermined plan regardless of opponent)
- **-1**: Erratic or contradictory ("I'll attack" → "I'll defend" → "I'll attack" without reason)

Dimension 3: Logical Coherence. Does later reasoning build causally on earlier conclusions?

- **+1**: Causal chain ("Because X, I did Y" → "Y resulted in Z, so next...")
- **0**: Independent but non-contradictory (each turn has reasoning but no cross-references)
- **-1**: Contradictory (earlier: "must defend," later: "should have attacked")

Special Cases:

- Trajectory with 1 to 2 turns: default score 0 (cannot evaluate evolution)
- Empty reasoning (`<think></think>`): score -1 (reasoning collapse)
- Very short reasoning (<20 tokens per turn): tend toward negative scores

Output JSON: `{"reasoning_deepening": <-1 to 1>, "strategy_adaptation": <-1 to 1>, "logical_coherence": <-1 to 1>, "explanation": "<50-100 words>"}`

Figure 16: Complete prompt template for computing the Reasoning Evolution Reward (ψ). The evaluator assesses three dimensions: reasoning deepening, strategy adaptation, and logical coherence.

(π_0^*, π_1^*) is a Nash equilibrium if neither player can improve by unilaterally deviating.

Alternating Turn Structure. In our formulation, players take turns rather than acting simultaneously. At turn t, only player $p = t \bmod 2$ acts, while the other player's action is null. This simplifies the transition dynamics:

$$s_{t+1} = T(s_t, a_t^{(p)}) \quad \text{where } p = t \bmod 2 \tag{8}$$

The alternating structure naturally models games like chess, Go, and the strategic games in our training suite (Tic-Tac-Toe, Kuhn Poker).

H Training Settings Details

This section provides complete hyperparameter configurations for reproducing our experiments.

Optimization Configuration. Training proceeds for 400 steps with 128 samples per step, yielding 51,200 game transitions total. We use Adam (Kingma and Ba, 2015) with learning rate 1×10^{-6}, batch size 128, and discount factor $\gamma = 1.0$. For role-conditioned advantage estimation, we set EMA decay $\alpha = 0.95$. Trajectories are sampled at temperature $\tau = 1.0$ to encourage exploration.

STRATAGEM-Specific Parameters. We set the Reasoning Evolution Reward coefficient $\beta = 0.2$ (Equation 3). The Reasoning Transferability Coefficient φ and Reasoning Evolution Reward ψ are computed using GPT-4 as the evaluation backbone, with prompts detailed in §D.2.

Computational Resources. All experiments run on 2 NVIDIA A100 GPUs (80GB) using a distributed actor-learner architecture. Actors generate self-play trajectories using vLLM (Kwon et al.,

Mathematical Reasoning Prompt

<|im_start|>user

Please reason step by step, and put your final answer within \boxed{}.

Question: {input}<|im_end|>

<|im_start|>assistant

Figure 17: Prompt template for mathematical reasoning benchmarks (MATH500, AIME, AMC, OlympiadBench, Minerva Math).

Multiple Choice Prompt

Please reason step by step, and put your final answer within \boxed{}. Your final answer should be of the following format: \boxed{LETTER} where LETTER is one of ABCD.
Question: {question}

Options:

A) {A}

B) {B}

C) {C}

D) {D}

Figure 18: Prompt template for multiple choice benchmarks (GPQA, MMLU-Pro). For MMLU-Pro, options extend to A through J.

2023) for efficient inference. Each full training run completes in approximately 30 hours.

I Game Environment Details

This section provides detailed descriptions of the three text-based zero-sum games used for training.

Tic-Tac-Toe. A classic 3×3 grid game serving as our testbed for *spatial reasoning*. Players alternate placing marks to form horizontal, vertical, or diagonal lines of three. The game requires pattern recognition, anticipating opponent moves, and multi-step forcing sequences. As a deterministic perfect-information game, Tic-Tac-Toe isolates pure strategic reasoning from uncertainty management.

Kuhn Poker. A simplified poker variant (Kuhn, 2016) emphasizing *probabilistic reasoning*. The game uses only three cards (Jack, Queen, King), where each player receives one card and must decide whether to bet, call, or fold based on incomplete information. Success demands probability estimation, opponent modeling, and expected value calculation under uncertainty.

Code Generation Prompt

Read the following function signature and docstring, and fully implement the function described. Your response should only contain the code for this function.

{function_signature_and_docstring}

Figure 19: Prompt template for code generation benchmark (HumanEval).

Simple Negotiation. A resource trading game developing *strategic optimization* skills. Two players exchange Wood and Gold tokens under opposing utility functions, creating natural tension between competing objectives. Players must infer opponent preferences, plan multi-step trade sequences, and communicate strategically through proposals.

J Out-of-Distribution Evaluation Games

We evaluate generalization to games never seen during training. Each OOD game is designed to test whether specific cognitive skills from training games transfer to novel mechanics.

Snake. A dynamic spatial reasoning game where two players control snakes on a grid, competing to collect apples while avoiding collisions with walls, themselves, or opponents. This tests whether static pattern recognition from Tic-Tac-Toe transfers to trajectory planning and dynamic obstacle avoidance in a real-time environment.

Pig Dice. A risk-reward decision making game where players repeatedly roll dice to accumulate points but lose all turn points when rolling 1. Players must decide when to "bank" accumulated points versus continuing to roll. This tests whether probabilistic reasoning from Kuhn Poker extends to sequential risk assessment and expected value calculation in different contexts.

Truth and Deception. An asymmetric information game where one player (the Deceiver) knows the true fact among several options and attempts to mislead through conversation, while the other player (the Guesser) must identify truth through strategic questioning. This evaluates whether negotiation skills transfer to pure communication strategy under information asymmetry.

K SPIRAL Framework Details

This section provides an extended introduction to SPIRAL (Liu et al., 2025), the self-play reinforce-

ment learning framework that serves as the foundation for our method.

K.1 Overview

SPIRAL (Self-Play on Zero-Sum Games Incentivizes Reasoning via Multi-Agent Multi-Turn Reinforcement Learning) trains language models through competitive self-play on strategic games. The key insight is that zero-sum games provide naturally verifiable rewards without requiring external annotators or reward models: a player either wins, loses, or draws, providing unambiguous training signal.

K.2 Self-Play Training Loop

SPIRAL's training proceeds as follows:

1. **Game Sampling**: Sample a game $G \sim \mathcal{G}$ from the game distribution
2. **Trajectory Generation**: Two instances of the current policy π_θ play against each other, generating trajectory $\tau = \{(s_t, y_t^{(p)})\}_{t=0}^{T}$
3. **Outcome Determination**: The game engine determines the winner, assigning rewards $R_p(\tau) \in \{-1, 0, +1\}$
4. **Policy Update**: Update θ using policy gradient with role-conditioned advantages

The self-play mechanism ensures automatic curriculum learning: as the policy improves, its opponent (itself) also improves, maintaining a challenging training distribution throughout learning.

K.3 Role-Conditioned Advantage Estimation

A critical challenge in two-player games is that the expected return differs by role. For example, in Tic-Tac-Toe, Player 0 (moving first) has structural advantage. Naively using the same baseline for both players leads to biased gradients.

SPIRAL addresses this through Role-conditioned Advantage Estimation (RAE), maintaining separate baselines $b_{G,p}$ for each game-role pair (G, p):

$$A_{G,p}(\tau) = R_p(\tau) - b_{G,p} \tag{9}$$

The baseline is updated via exponential moving average:

$$b_{G,p} \leftarrow \alpha \cdot b_{G,p} + (1 - \alpha) \cdot R_p(\tau) \tag{10}$$

where α is the smoothing coefficient (typically 0.99).

K.4 Policy Gradient Formulation

The policy gradient for SPIRAL aggregates over all turns played by each role:

$$\nabla_\theta J = \mathbb{E}_{G,\tau}\left[\sum_{p\in\{0,1\}}\sum_{t\in\mathcal{T}_p} A_{G,p}(\tau)\nabla_\theta \log \pi_\theta(y_t^{(p)}|s_t, p, G)\right] \tag{11}$$

where $\mathcal{T}_p = \{t : t \bmod 2 = p\}$ indexes the turns belonging to player p.

The role conditioning is implemented by prepending a role identifier to the prompt, enabling a single policy to model both players' behavior while accounting for role-specific strategic considerations.

K.5 Limitations and Motivation for STRATAGEM

While SPIRAL demonstrates that game-based self-play can improve reasoning, transferring these capabilities to domains like mathematics and coding faces two fundamental challenges:

1. **Domain Specificity**: SPIRAL optimizes for game outcomes without explicitly encouraging abstract reasoning patterns. Winning strategies often rely on game-specific heuristics (e.g., "King beats Queen") rather than domain-agnostic patterns (e.g., "enumerate cases and compute expected value").
2. **Contextual Stasis**: Games present static problem contexts where rules and settings remain fixed throughout interaction. SPIRAL does not incentivize reasoning that adapts to evolving contexts, yet real-world problems (e.g., mathematical proofs, code debugging) require continuous adaptation as intermediate results reshape the solution space.

These challenges fundamentally limit reasoning transfer. To incentivize transferable reasoning, STRATAGEM addresses both challenges through trajectory advantage modulation: φ overcomes domain specificity by measuring abstraction level, while ψ overcomes contextual stasis by rewarding adaptive reasoning development.