WHAT-IF ANALYSIS OF LLMS: EXPLORE THE GAME WORLD USING PROACTIVE THINKING

Yuan Sui[1]*, Yanming Zhang[2], Yi Liao[3], Yu Gu[3], Guohua Tang[3], Zhongqian Sun[3] Wei Yang[3], Bryan Hooi[1]
[1]National University of Singapore, [2]Zhejiang University, [3]Tencent

ABSTRACT

LLMs struggle with decision-making in high-stakes environments such as MOBA games, primarily due to limited proactive reasoning and an incomplete understanding of complex game dynamics. To address this, we propose What-if Analysis LLM (WiA-LLM), a framework that trains an LLM as an explicit, language-based world model. Instead of representing the environment in latent vectors, WiA-LLM uses natural language to simulate how the game state evolves over time in response to candidate actions and provides textual justifications for these predicted outcomes. WiA-LLM is trained in two stages: supervised fine-tuning on human-like reasoning traces, followed by reinforcement learning with outcome-based rewards that align predicted and actual future states. In the Honor of Kings (HoK) environment, WiA-LLM attains 74.2% accuracy (27%↑ vs. the base model) in forecasting game-state changes. In addition, WiA-LLM demonstrates strategic behavior more closely aligned with expert players than purely reactive LLMs, indicating enhanced foresight and expert-like decision-making.

1 INTRODUCTION

What-If Analysis (WIA), as the name suggests, is a systematic decision-making approach that addresses hypothetical questions such as "*What if we take this action? How will it affect the final outcome?*". WIA enables decision-makers to simulate hypothetical scenarios by altering specific action variables and assessing their potential implications Gathani et al. (2022). This methodology is valuable for strategic planning, risk assessment, and enhancing the explainability of the decision-making process Tang et al. (2025).

Despite its potential, the integration of WIA capabilities into large language models (LLMs) remains underexplored. This reveals a critical limitation of current LLMs when applied to dynamic, high-stakes scenarios such as strategic planning, risk assessment, and real-time decision-making. While existing LLMs excel at *reactive thinking* Liao et al. (2025); Wei et al. (2023); Li et al. (2025)—generating answers based on the current context and their prior knowledge—they lack mechanisms for *proactive thinking*, which is essential for forecasting the consequences of actions before they occur. This limitation is particularly pronounced in dynamic environments, where each action may trigger a series of cascading effects Wei et al. (2022); Hu et al. (2024c); Sui et al. (2025). Understanding the consequences of different actions not only clarifies the environment but also provides deeper intuition for making informed decisions.

To address this gap, we introduce **WiA-LLM**, a novel framework that endows LLMs with proactive thinking capabilities through explicit world modeling. By leveraging environmental feedback via reinforcement learning (RL), WiA-LLM learns to forecast the outcomes of different actions on the entire game state. Our core insight draws from human cognition: **"Look before you leap"**, i.e., one should consider possible consequences or dangers before acting. We formalize this process as explicit world modeling: $S_\Delta = f(S_t, a_t)$, where the model predicts the state transition S_Δ resulting from taking action a_t in state S_t using natural language. The task becomes increasingly challenging as more properties change in S_Δ (see §3.2). Our training pipeline first applies supervised fine-tuning (SFT) on human gameplay trajectories to provide basic behavioral and environmental knowledge. It

*Corresponding Email: `yuan.sui@u.nus.edu`

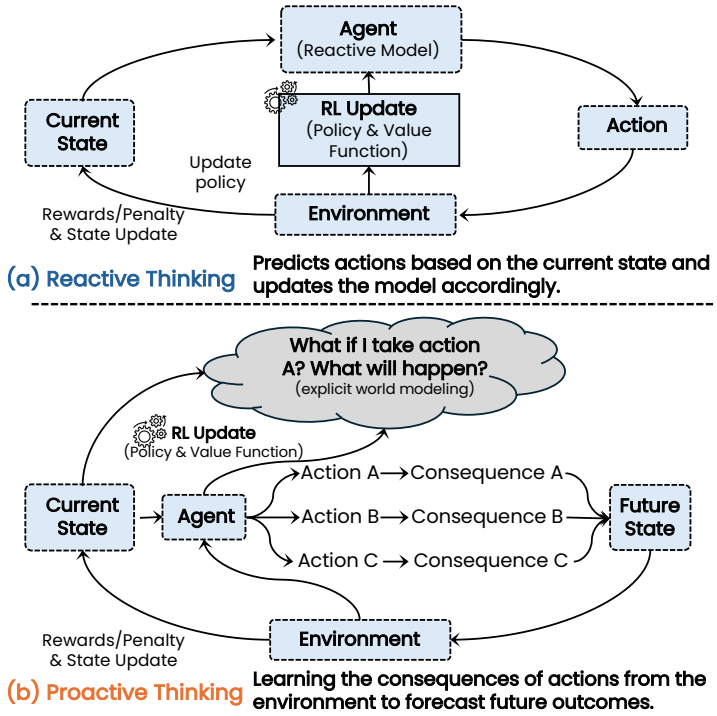

Figure 1: **Illustration of reasoning paradigms**: (a) reactive thinking, where the model selects an action given the current game state; (b) proactive thinking, where the model also forecasts the consequences of candidate actions on future game states. In this work, we focus on proactive thinking and train models to forecast the consequences of different actions.

then applies RL with rule-based, verifiable rewards that compare the model's forecasts against actual environment transitions, aligning its predictions with real dynamics (see §3.3). In this way, WiA-LLM shifts LLMs from purely reactive pattern matching to model-based forecasting, analogous to how humans mentally simulate outcomes before action.

We evaluate WiA-LLM in *Honor of Kings* (HoK), a large-scale multiplayer online battle arena (MOBA) game Wei et al. (2022). HoK serves as an ideal testbed for three reasons. First, it exhibits *high dynamic complexity*: players must adapt in real time to over one hundred heroes, shifting objectives, and coordinated team strategies. Second, it offers *quantifiable states*: the game state can be encoded as JSON-structured objects with hero positions, resources, and map conditions, enabling precise and automatic reward computation. Third, it features *high-stakes consequences*: a single mistimed objective contest (e.g., a dragon fight) can flip the match outcome, creating a rich space for WIA evaluation. To thoroughly test whether WiA-LLM can support dynamic proactive reasoning and adapt decision-making based on forecasted consequences, we construct two benchmarks in this environment (see §3.2 and §4.1).

Our experiments demonstrate that: (1) WiA-LLM achieves 74.2% forecasting accuracy on HoK scenarios, outperforming the base model (qwen3-14b) by 27% and surpassing deepseek-r1 by 41.6%; (2) it enhances downstream decision-making, enabling agents to exhibit strategic behavior closer to that of expert human players; (3) it consistently improves across difficulty tiers, achieving 93.9% accuracy on the simplest forecasting tasks and 73.1% on moderately complex tasks; (4) it demonstrates stable optimization during RL training, with reward convergence within 400 steps and no observed degradation in output quality; and (5) it maintains strong zero-shot generalization on standard academic benchmarks. **Overall, our contributions are as follows:**

- We propose WiA-LLM, a framework that performs proactive thinking by forecasting the consequences of actions using environmental feedback.
- We design a training paradigm that combines SFT on human gameplay data with RL guided by rule-based, verifiable rewards, aligning model forecasts with actual environment transitions.

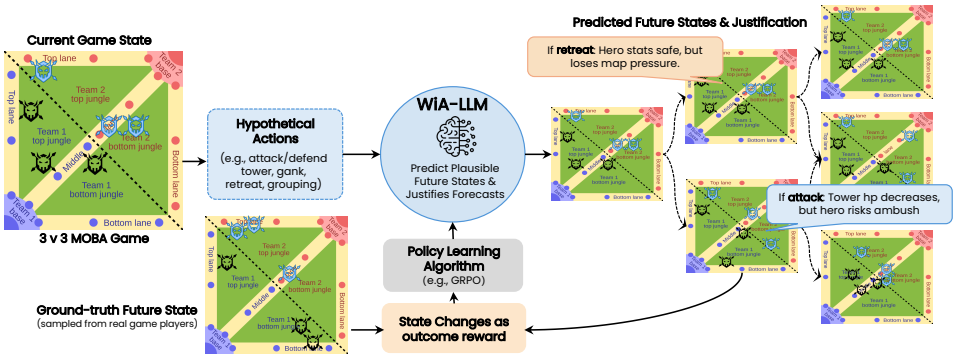

Figure 2: **Workflow** of WiA-LLM. Given the current game state and a set of hypothetical actions, the model is tasked with forecasting the potential changes to the entire game state that would result from each action, and providing justifications for these forecasts. The predicted game state changes are then compared to ground-truth values using a rule-based verifier, which is used to update the policy model. This process enables the model to perform what-if analysis by simulating action outcomes and iteratively refining its decision-making.

- We demonstrate that WiA-LLM achieves strong performance in a complex MOBA environment and enables agents to exhibit strategic behavior closer to expert human players.

2 Definition

In this section, we formally define the What-If Analysis task within the game environment and introduce the key notation used in our framework.

2.1 Task Definition: What-If Analysis

Given a game state S_t at time step t, the objective is to forecast the resulting state change, denoted as S_Δ, after the player executes a specific action a_t. This task requires the model to reason about the current environmental conditions and demonstrate a deep understanding of game mechanics to predict the causal impact of each action. As illustrated in Figure 1, WIA reflects a shift toward *proactive thinking*: instead of simply reacting to the current state, the model anticipates the future consequences of its decisions, thus enabling more informed strategic planning.

2.2 Task Definition: Action Prediction

We consider action prediction as one of the downstream tasks in game decision-making. This task can be formalized as $a_t^* = f(S_t)$, where the model predicts the optimal action based on the current game state S_t. In the HoK environment, the available actions are defined as a set A (see Table D for details). Notably, these actions are designed at the strategic level, providing high-level guidance to users who then adjust their low-level operations (such as moving the hero or using abilities) accordingly. This work differs from prior studies Wei et al. (2022); Qu et al. (2023), which primarily focus on low-level actions like changing hero positions and skill releases.

2.3 Game State

We model the game environment as a sequence of discrete states. Each state S_t encapsulates all visible information from the player's perspective, including teammate attributes, visible turrets, and map vision. To ensure realistic gameplay, we strictly enforce **partial observability** by excluding hidden information, such as the status of enemies concealed by the "fog of war." For compatibility with LLMs, the game state S_t is serialized into a JSON object—a format that LLMs can naturally parse and process.

2.4 GAME ENVIRONMENT

We conduct our study in *Honor of Kings* (HoK) Wei et al. (2022); Qu et al. (2023). In HoK, players control unique heroes and coordinate with teammates to defeat opponents, neutral creatures, and defensive structures, with the ultimate goal of destroying the opposing team's base crystal. HoK serves as an ideal testbed due to its complexity: the need for team coordination, dynamic strategic shifts, and the high-dimensional state space present significant challenges for decision-making. Mastering proactive reasoning in such a complex domain holds great promise for advancing the reasoning capabilities of game AI.

3 METHOD: WIA-LLM

3.1 OVERVIEW & PROBLEM FORMULATION

Our framework, WiA-LLM, facilitates the transition of LLMs from reactive pattern matching to proactive world modeling. We conceptualize the What-If Analysis task as a conditional state prediction problem within a partially observable Markov Decision Process (POMDP) Wikipedia (2025). Formally, let $\mathcal{S}$ and $\mathcal{A}$ denote the game state and action space. At time step t, the agent observes a state S_t and chooses a candidate action $a_t \in \mathcal{A}$. The objective is to approximate the transition function $\mathcal{T}(S_{t+1}|S_t, a_t)$ by predicting the future state changes S_Δ, where S_Δ represents the causal impact of the action on the entire state. Unlike standard next-token prediction, WiA requires the model to implicitly simulate the environment dynamics and generate justifications for the state transition.

3.2 DATA CONSTRUCTION AND UTILIZATION

To ground the model in realistic dynamics, we construct a scalable dataset derived from *Honor of Kings (HoK)*. This dataset is pivotal for our multi-stage training pipeline, providing both the reasoning priors for SFT and the ground-truth oracle for RL verification.

Trajectory Parsing and Difference Extraction. We process raw gameplay logs using a strict state-parsing pipeline (Algorithm 1) to generate the base transition dataset $\mathcal{G}_D = \{(S_t, a_t, S^*_\Delta)_i\}_{i=1}^N$. Specifically, each state S_t is first serialized into a structured JSON object encompassing all visible information, such as hero attributes, turret status, and map vision, to enforce partial observability. We then parse the player's executed action a_t corresponding to S_t using a predefined taxonomy. To compute the target label, we calculate the ground-truth state difference S^*_Δ by comparing S_t with the future state $S_{t+\delta}$ (where $\delta > 0$). This difference vector S^*_Δ captures changes in critical components $\mathcal{C} = \{\text{hero, tower, minions, dragon}\}$ and serves as the definitive label for environmental consequences. The gathered dataset $\mathcal{G}_D$ is utilized across the two optimization stages described below.

Stage I: Reasoning Distillation (SFT). To endow the model with strong reasoning capabilities, we augment the base training samples with synthetic reasoning traces. Leveraging a teacher model (DeepSeek-R1) that has access to the ground truth (S_t, a_t, S^*_Δ), we distill a "thinking process" C_t that explains the causal link between the action and its outcome. This yields the SFT corpus $\mathcal{D}_{SFT} = \{(S_t, a_t, C_t, S^*_\Delta)\}$, which is used to train the model to generate structured reasoning before producing the final prediction.

Stage II: Outcome Verification (RL). For the reinforcement learning stage, we revert to the original real-world data $\mathcal{G}_D$. Here, the pair (S_t, a_t) serves as the prompt q, while the ground truth S^*_Δ is reserved as the oracle for the rule-based reward function. This ensures that the policy optimization is driven by actual environmental dynamics rather than distilled approximations.

3.3 POLICY LEARNING VIA VERIFIABLE REWARDS

While SFT provides basic behavioral patterns, it lacks the ability to effectively guide the model to self-explore in dynamic environments. We therefore employ Group Relative Policy Optimization (GRPO Shao et al. (2024)), which directly aligns the policy's forecasts with actual environmental transitions. Following the success of Deepseek-R1 DeepSeek-AI et al. (2025), we adopt a similar

rule-based reward r_t to measure the alignment between the predicted state change S_Δ and the ground truth S^*_Δ:

$$r_t = \frac{\sum_{(k,v_k)\in S_\Delta} w_k \cdot \text{Score}(v_k, v^*_k)}{\sum_{(k,v)\in S_\Delta} w_k} \tag{1}$$

where w_k denotes the weight assigned to each key k, reflecting its relative importance. The scoring function assigns a value of 1 for exact matches, 0.5 for partial matches, and 0 otherwise. This reward encourages the model to generate action predictions that closely match real player behavior while penalizing overly verbose or irrelevant outputs. We do not incorporate format rewards, as our learned model already demonstrates strong structural adherence. Detailed formulation is provided in Appendix A, and training prompts can be found in the Appendix D.

3.4 Inference-Time Decision Making

To leverage the learned world model for strategic gameplay, we implement a **lookahead search mechanism** during inference (see Figure 7). Unlike reactive agents that map states directly to actions ($\pi : S \to a$), our approach explicitly reasons about future outcomes before committing to a decision. Specifically, the agent performs a one-step lookahead search:

- **Candidate Generation:** Given the current state S_t, the model first acts as a policy proposal network, sampling a set of k plausible candidate actions $\mathcal{A}_{cand} = \{a_1, ..., a_k\}$ from the policy distribution $\pi_\theta(\cdot|S_t)$.
- **What-if Simulation:** For each candidate action $a_i \in \mathcal{A}_{cand}$, WiA-LLM is queried to forecast the corresponding differential state transition $S_{\Delta,i}$. This step represents the core "What-If" analysis, where the model conditions on the hypothetical execution of a_i to produce $S_{\Delta,i} = f_\theta(S_t, a_i)$.
- **Heuristic Evaluation:** To ensure robust selection, we employ a deterministic, rule-based value function $V(S_\Delta)$. This function acts as a classifier, categorizing the forecasted consequences (e.g., tower destroyed, resource gained, hero death) into strategic values (positive or negative).
- **Optimal Selection:** The final action a^* is selected by maximizing the evaluated outcome of the simulated future: $a^* = \text{argmax}_{a_i\in\mathcal{A}_{cand}} V(S_{\Delta,i})$. This decouples the generation of possibilities from the evaluation of strategic success, mitigating the risk of the model rationalizing suboptimal plans.

This "simulate-and-evaluate" is the essence of model-based planning. It allows the agent to "think ahead" and select the action with the best forecasted outcome, rather than simply choosing the action that seems best in a reactive manner. However, we acknowledge that LLM inference introduces a substantial latency, making it impractical to run this process on every game frame. To address the latency issue, we employ a dual-system architecture (see Appendix B), where WiA-LLM acts as a low-frequency strategic planner (e.g., every 5-10s) guiding a high-frequency reactive policy.

4 Experiments

4.1 Experiment Setup

Environment. All experiments were conducted on four servers, each equipped with 8 NVIDIA H20 GPUs (96 GB each). For SFT, we used the Megatron-LM Shoeybi et al. (2019) training platform, while online RL was performed using OpenRLHF Hu et al. (2024a).

Datasets. We curate two new benchmarks for WIA tasks: **WIA-General** and **WIA-Hardest**. Task difficulty is quantified by the number of altered game-critical components, denoted as $d = \|S_\Delta\|$, where S_Δ is the set of differences drawn from $\mathcal{C} = \{\text{hero}, \text{tower}, \text{minion_waves}, \text{dragon}\}$ as defined in Algorithm 1. The difficulty d ranges from 1 (a single-component change) to 4 (simultaneous changes to all components). The statistics for WIA-General ($\mathcal{D}_g = \{(S_i, a_i, S_\Delta) \mid 1 \le d \le 4\}$) and WIA-Hardest ($\mathcal{D}_h = \{(S_i, a_i, S_\Delta) \mid d = 4\}$) are provided in Table 1, with results presented in Table 2. We also verify that domain-specific training does not degrade general capabilities; see Appendix C for details.

Baselines. We use LLMs of various scales as baselines, primarily focusing on the Qwen3 models Yang et al. (2025) with native reasoning capabilities, including Qwen3-14B and Qwen3-8B. We

Metric	WIA-General (n=1,476)		WIA-Hardest (n=452)		Combined (n=1,928)	
	Count	%	Count	%	Count	%
Change Types Distribution						
Minion Changes	1,003	67.95	452	100.00	1,455	75.47
Tower Changes	1,355	91.80	452	100.00	1,807	93.73
Hero Changes	412	27.91	452	100.00	864	44.81
Dragon Changes	37	2.51	452	100.00	489	25.36
Difficulty Levels Distribution						
d=1 (1 change)	524	35.50	0	0.00	524	27.18
d=2 (2 changes)	567	38.41	0	0.00	567	29.42
d=3 (3 changes)	369	25.00	0	0.00	369	19.15
d=4 (4 changes)	16	1.08	452	100.00	468	24.28

Table 1: Statistics: WIA-General & WIA-Hardest.

Model	Overall Benchmarks		Difficulty Breakdown (d)			
	WIA-General	WIA-Hardest	$d = 1$	$d = 2$	$d = 3$	$d = 4$
deepseek-r1	0.326	0.111	0.443	0.298	0.213	0.102
deepseek-r1-distilled-qwen-3-14B	0.370	0.046	0.566	0.339	0.156	0.023
qwq-32b	0.366	0.037	0.467	0.361	0.246	0.016
qwen-3-14b	0.472	0.027	0.640	0.486	0.234	0.008
WiA-LLM (14b-sft)	0.614	0.281	0.777	0.590	0.433	0.297
WiA-LLM (14b-grpo)	0.674	0.132	0.895	0.660	0.404	0.117
WiA-LLM (14b-sft-grpo) (our best model)	**0.742**	**0.295**	**0.939**	**0.731**	**0.497**	**0.312**
qwen-3-8b	0.450	0.022	0.601	0.457	0.245	0.023
WiA-LLM (8b-sft)	0.669	0.179	0.898	0.625	0.431	0.172
WiA-LLM (8b-grpo)	0.619	0.108	0.831	0.601	0.367	0.094
WiA-LLM (8b-sft-grpo) (our best model)	**0.742**	**0.426**	**0.938**	**0.728**	**0.500**	**0.430**

Table 2: Combined performance comparison. The left columns show results on the WIA-General and WIA-Hardest benchmarks. The right columns detail model performance across specific difficulty levels ($d = 1$ to $d = 4$). Our best models (*-sft-grpo) consistently outperform baselines across all metrics.

also consider several other models: DS-R1, DS-R1-distilled-Qwen3-14B, and QwQ-32B. All model checkpoints are accessible via Hugging Face.

4.2 Training Details

Building on insights from Deepseek-R1 DeepSeek-AI et al. (2025), we employ a multi-stage training strategy that combines supervised fine-tuning (SFT) and reinforcement learning (RL) to enhance the capabilities of our language models. Specifically, SFT improves the foundational language understanding and reasoning abilities of our models, while online RL enables efficient exploration and selection of the most effective solutions through trial and error.

We select Qwen3-8B and Qwen3-14B as our base models. For the SFT stage, we curate the training dataset by distilling knowledge from Deepseek-R1, which demonstrates strong reasoning capabilities in game environments and can thoroughly analyze game states based on its pre-existing knowledge. Specifically, we provide the ground truth game state changes and the corresponding instruction to the R1 model, allowing it to generate a reasoning process Wei et al. (2023) that leads to the final answer. The prompt used for this process is provided in Appendix D. The distilled data are formatted as (game state: S_t, action: a_t, and thinking process: C_t) and serve as a valuable resource for training smaller models to acquire R1-like deep reasoning skills. For the online RL stage, we use real gameplay data collected as described in Section 3. We explore two setups: (1) applying GRPO directly to the base model without any prior SFT and (2) applying GRPO to a model that has already been fine-tuned. These two setups are compared in Table 2. Due to computational constraints, we

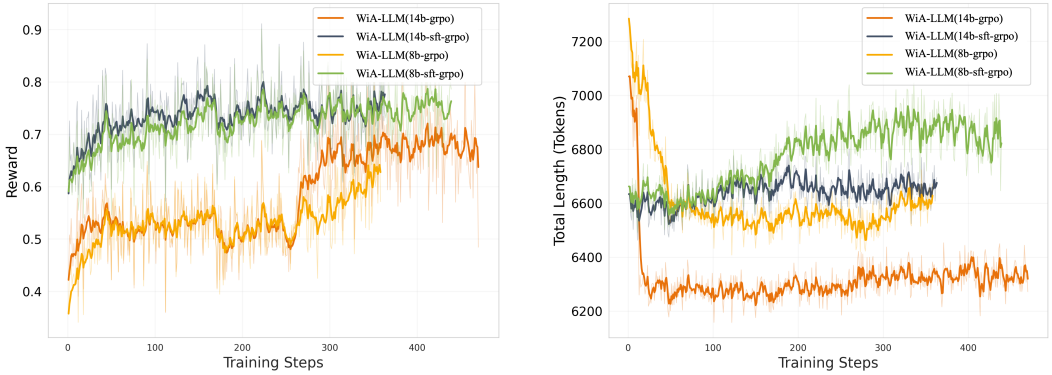

Figure 3: Demonstration of reward progression (left) and total token length (right) over training steps during the RL process. The results show that our method consistently achieves higher rewards and maintains more stable or longer token lengths compared to the baselines, indicating improved learning efficiency and output quality.

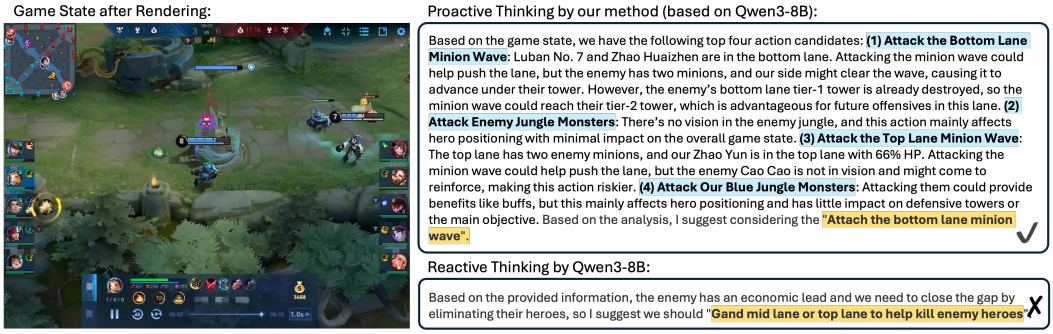

Figure 4: Case study on WiA. To safeguard user privacy, we blurred the user's Game ID.

limit GRPO training to approximately 400 steps for all models to ensure a fair comparison, and set the number of epochs for the SFT stage to three.

4.3 Main Results

The main results are presented in Table 2. These results reveal several critical insights into the effectiveness of our multi-stage training approach. Our WiA-LLM with SFT+GRPO consistently achieves superior performance across all evaluation settings, with particularly notable improvements on the challenging subsets (WIA-Hardest). On WIA-General, both WiA-LLM-14B and WiA-LLM-8B with SFT+GRPO attain nearly identical performance (0.742), substantially outperforming a much larger model such as Deepseek-R1 (0.326). The performance gain is even more pronounced on WIA-Hardest, where our 8B model achieves 0.426 accuracy compared to Deepseek-R1's 0.111, while the 14B model reaches 0.295. We further evaluate performance across difficulty levels on WIA-General. While all models experience degradation as task complexity increases from $d = 1$ to $d = 4$, our approach maintains the most robust performance. Notably, on the most challenging $d = 4$ task, our 8B model achieves 0.430 accuracy (compared to Deepseek-R1's 0.102), and our 14B model reaches 0.312, which remarkably exceeds the 14B baseline's performance at this difficulty level. These results demonstrate that both SFT and GRPO independently deliver substantial performance improvements; however, when combined, they yield even greater gains, surpassing the results of either method alone.

We also verified that domain-specific training did not degrade general capabilities; see Appendix C for results on MMLU/Math/BBB and other benchmarks.

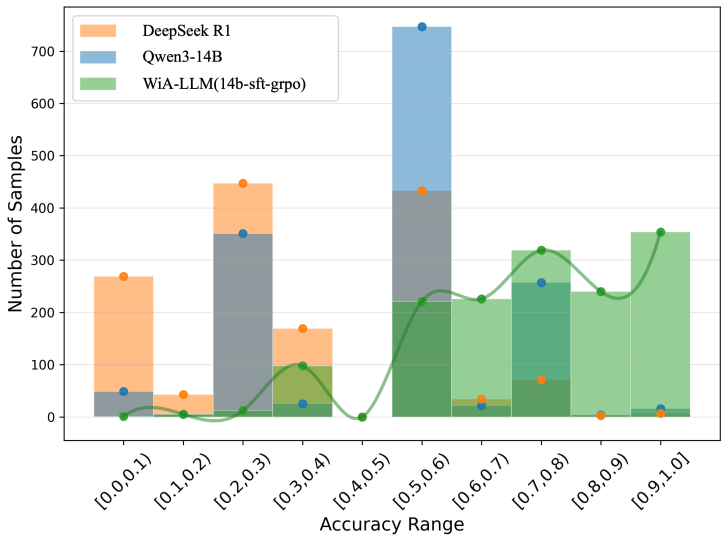

Figure 5: Distribution of sample counts across different accuracy ranges. The stacked histograms illustrate the number of samples per accuracy interval, while the smoothed trend line highlights the performance pattern of our method.

4.4 Analysis

Response Length vs. Rewards. Figure 3 shows that all models achieve consistent reward improvements, with SFT-initialized variants (WiA-LLM-*-sft-grpo) starting from significantly higher baselines (0.6 vs. 0.35). While reward trajectories remain largely consistent across models, response lengths differ: SFT models maintain stable or growing output lengths, whereas non-SFT models exhibit initial fluctuations before stabilizing. This suggests that reasoning distillation (SFT) helps models better balance the elaboration of reasoning with reward optimization during RL.

Distribution of Performance Range. To further validate the robustness of our model's performance—and to avoid relying solely on aggregate metrics—we examine the distribution of sample counts across different accuracy ranges. As shown in Figure 5, we plot the sample counts for various models over these ranges and include a smoothed trend line that highlights the performance pattern of WiA-LLM(14b-sft-grpo). The results demonstrate that our model outperforms baselines by concentrating a larger proportion of samples in the higher accuracy intervals (0.7 to 1.0), indicating more consistent and accurate predictions compared to the base models.

Case Studies. To assess downstream utility, we conduct a detailed case study illustrated in Figure 4. We apply the lookahead search strategy (see Section 3.4) to the action prediction task (defined in Section 2). We find that WiA-LLM produces more detailed and verifiable reasoning traces for each action, and accurately simulates minion wave mechanics to justify a strategic lane push. In contrast, the reactive baseline (Qwen3-8b) hallucinates a non-existent ganking opportunity, underscoring the advantages of proactive reasoning.

5 Related Works

Game Understanding of LLMs. While LLMs excel at language-based reasoning, applying them effectively to games remains challenging due to their reliance on static pre-training data and a lack of environmental grounding Hu et al. (2024b). Key challenges include: (1) *Contextual grounding*—difficulty in interpreting dynamic game states for consistent decision-making Hu et al. (2024c); (2) *Symbolic precision*—misinterpretation of game terminology and item attributes, which can disrupt interaction with the game engine Southey et al. (2012); and (3) *Long-term planning*—limited memory and strategic reasoning over extended horizons Silver et al. (2016); Vinyals et al. (2017); He et al. (2025).

Role of RL in LLMs. Recent advances in LLMs have highlighted the important role of RL in aligning model outputs with human preferences Sui et al. (2025); Jin et al. (2025). While pre-training on large text corpora enables LLMs to generate fluent and grammatically correct text, this alone does not guarantee that models are helpful, harmless, or aligned with user expectations. RL from human feedback (RLHF) Ouyang et al. (2022) addresses this by training a reward model based on human preferences to guide policy optimization via methods such as PPO Schulman et al. (2017), DPO Rafailov et al. (2023), and SimPO Meng et al. (2024). More recently, GRPO Shao et al. (2024) has emerged as a flexible alternative for obtaining reward signals. Unlike PPO, GRPO does not strictly require a reward model; instead, it can incorporate reward signals from any function or model capable of evaluating response quality.

Difference from Time-Series Forecasting. Time Series Forecasting (TSF) is a related task that focuses on predicting future values of target variables based on historical data patterns Tang et al. (2025); Shi et al. (2022). Its goal is to extrapolate trends or behaviors from past observations to future time steps Jin et al. (2024). In contrast, WIA is concerned with understanding the causal impact of specific actions or interventions on the future state of the environment. Rather than simply predicting what will happen next, WIA actively evaluates multiple candidate actions to determine which choice leads to the most beneficial outcome. In other words, TSF forecasts the natural progression of a system, while WIA simulates hypothetical scenarios to inform decision-making by assessing the consequences of different actions. This capability is especially valuable for complex decision-making in dynamic environments.

6 Conclusion

We propose WiA-LLM as an explicit world model that enables LLMs to proactively forecast the consequences of actions. Through interaction with game environments, WiA-LLM develops a deeper understanding of state dynamics and improves decision-making. While evaluated in *Honor of Kings*, our approach is broadly applicable to other high-stakes domains where simulating outcomes is safer than trial and error.

References

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Sneha Gathani, Madelon Hulsebos, James Gale, Peter J. Haas, and Çağatay Demiralp. Augmenting decision making via interactive what-if analysis, 2022. URL https://arxiv.org/abs/2109.06160.

Yufei He, Ruoyu Li, Alex Chen, Yue Liu, Yulin Chen, Yuan Sui, Cheng Chen, Yi Zhu, Luca Luo, Frank Yang, and Bryan Hooi. Enabling self-improving agents to learn at test time with human-in-the-loop guidance. *arXiv preprint arXiv: 2507.17131*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024a.

Sihao Hu, Tiansheng Huang, Gaowen Liu, Ramana Rao Kompella, Fatih Ilhan, Selim Furkan Tekin, Yichang Xu, Zachary Yahn, and Ling Liu. A survey on large language model-based game agents. *arXiv preprint arXiv: 2404.02039*, 2024b.

Sihao Hu, Tiansheng Huang, and Ling Liu. Pokellmon: A human-parity agent for pokemon battles with large language models, 2024c. URL https://arxiv.org/abs/2402.01118.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, jiayi lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 62991–63010. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c6ec1844bec96d6d32ae95ae694e23d8-Paper-Datasets_and_Benchmarks.pdf.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv: 2503.09516*, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

lanhin. School chinese benchmark, 2018. URL https://github.com/lanhin/SchoolChinese.

Junlong Li, Daya Guo, Dejian Yang, Runxin Xu, Yu Wu, and Junxian He. Codei/o: Condensing reasoning patterns via code input-output prediction. *arXiv preprint arXiv: 2502.07316*, 2025.

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

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, et al. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *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*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Yun Qu, Boyuan Wang, Jianzhun Shao, Yuhang Jiang, Chen Chen, Zhenbin Ye, Liu Linc, Yang Feng, Lin Lai, Hongyang Qin, Minwen Deng, Juchao Zhuo, Ye, et al. Hokoff: Real game dataset from honor of kings and its offline reinforcement learning benchmarks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 22166–22190. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/464fefa022aaefc85d901317bbf13f85-Paper-Datasets_and_Benchmarks.pdf.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv: 1707.06347*, 2017.

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

Jimeng Shi, Mahek Jain, and Giri Narasimhan. Time series forecasting (tsf) using various deep learning models, 2022. URL https://arxiv.org/abs/2204.11115.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv: 1909.08053*, 2019.

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

Finnegan Southey, Michael P Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. *arXiv preprint arXiv:1207.1411*, 2012.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=uyTL5Bvosj.

Yuan Sui, Yufei He, Tri Cao, Simeng Han, Yulin Chen, and Bryan Hooi. Meta-reasoner: Dynamic guidance for optimized inference-time reasoning in large language models, 2025. URL https://arxiv.org/abs/2502.19918.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.824. URL https://doi.org/10.18653/v1/2023.findings-acl.824.

Hua Tang, Chong Zhang, Mingyu Jin, Qinkai Yu, Zhenting Wang, Xiaobo Jin, Yongfeng Zhang, and Mengnan Du. Time series forecasting with llms: Understanding and enhancing model capabilities. *ACM SIGKDD Explorations Newsletter*, 26(2):109–118, 2025.

Quan Tu, Shilong Fan, Zihang Tian, and Rui Yan. Charactereval: A chinese benchmark for role-playing conversational agent evaluation. *arXiv preprint arXiv: 2401.01275*, 2024.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, et al. Starcraft ii: A new challenge for reinforcement learning, 2017. URL https://arxiv.org/abs/1708.04782.

Hua Wei, Jingxiao Chen, Xiyang Ji, Hongyang Qin, Minwen Deng, Siqin Li, Liang Wang, Weinan Zhang, Yong Yu, Liu Lin, et al. Honor of kings arena: an environment for generalization in competitive reinforcement learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *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*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets_and_Benchmarks.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Wikipedia. Partially observable Markov decision process — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Partially%20observable%20Markov%20decision%20process&oldid=1318763884, 2025. [Online; accessed 03-December-2025].

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

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. Ape210k: A large-scale and template-rich dataset of math word problems, 2020. URL https://arxiv.org/abs/2009.11506.

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

Algorithm 1 Game State Parsing and Difference Extraction

Require: Raw Game State Sequence $S = [S_1, \ldots, S_T]$
Ensure: Transition Dataset G_D
Constant $\mathcal{C} \leftarrow \{\text{hero}, \text{tower}, \text{minions}, \text{dragon}\}$
Initialize lists: $P \leftarrow [\,]$; $G_D \leftarrow [\,]$
for $t \leftarrow 1$ **to** T **do** ▷ *Phase 1: Annotate Actions*
    $a_t \leftarrow \textsc{Annotate}(S_t)$
    $P.\textsc{Append}((S_t, a_t))$
end for
for $i \leftarrow 1$ **to** $|P| - 1$ **do** ▷ *Phase 2: Extract Transitions*
    Let $(S_{cur}, a_{cur}) = P[i]$ and $(S_{next}, a_{next}) = P[i+1]$
    if $\textsc{TimeDelta}(P[i], P[i+1]) > 60\text{s}$ **then**
        continue
    end if
    if $a_{cur} \neq a_{next}$ **then**
        $S_\Delta \leftarrow \textsc{ComputeStateDiff}(S_{cur}, S_{next})$
        $G_D.\textsc{Append}((S_{cur}, a_{cur}, S_\Delta))$
    end if
end for
return G_D
function $\textsc{ComputeStateDiff}(S_{\text{old}}, S_{\text{new}})$
    $S_\Delta \leftarrow \text{Map}()$
    for each $c \in \mathcal{C}$ **do**
        if $S_{\text{old}}[c] \neq S_{\text{new}}[c]$ **then**
            $S_\Delta[c] \leftarrow \textsc{Diff}(S_{\text{old}}[c], S_{\text{new}}[c])$
        end if
    end for
    return S_Δ
end function

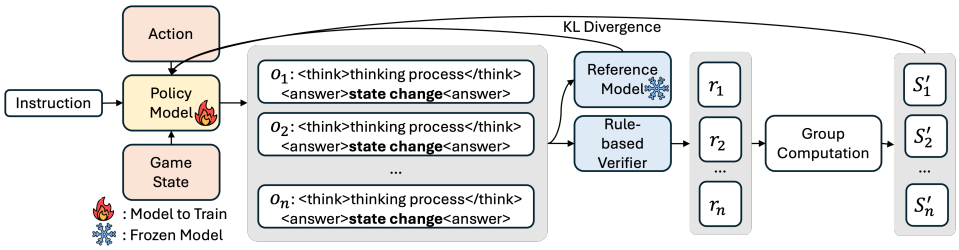

Figure 6: **Demonstration** of GRPO training with Game State. Given the player's action and the current game state, the model is asked to forecast the potential changes to the entire game state once the player takes the action, and provide the thinking process as the analysis of this what-if scenario. We then use the predicted game state changes to compare with ground-truth values using a rule-based verifier to update the policy model. This process enables the model to perform what-if analysis (forecasting) by simulating action outcomes and refining its decision-making accordingly.

LIMITATIONS

In this work, we focus on the *strategic reasoning capabilities* of LLMs. While we demonstrate superior alignment with expert human actions and high-fidelity state forecasting, we do not test our method in an online ranked environment to measure win rate. Online performance depends heavily on low-level execution (micro-mechanics and reaction time), which is outside the scope of this work on proactive world modeling.

A GRPO FORMULATION WITH GAME STATE

To facilitate effective learning of proactive reasoning in game environments, we employ *Group Relative Policy Optimization* (GRPO) Shao et al. (2024), an online RL algorithm designed to maxi-

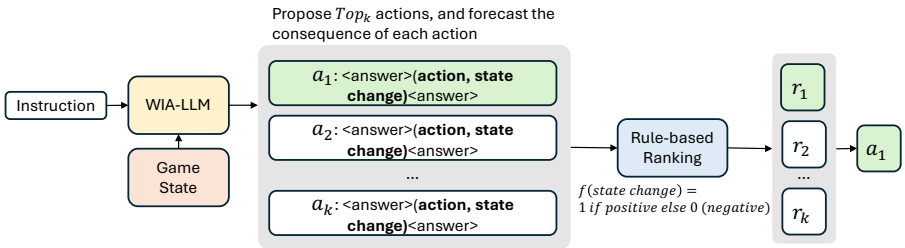

Figure 7: **Demonstration**: Adaptation of WiA-LLM to the downstream action prediction task.

mize the advantage of generated completions while constraining policy divergence from a reference model.

We formalize the training process of WiA-LLM using GRPO as follows. Let q denote a sampled prompt (e.g., a game state s_t and context i_t), and let $\{o_1, o_2, \ldots, o_G\}$ represent a group of G completions generated by the old policy $\pi_{\theta_{\text{old}}}$. For each completion o_i, a reward r_i is computed using a rule-based reward function (see Section 3.3). The group-relative advantage for each completion is then calculated as:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(r)}{\text{std}(r)}, \tag{2}$$

where $\text{mean}(r)$ and $\text{std}(r)$ denote the mean and standard deviation of rewards within the group, respectively. This normalization ensures the advantage reflects the relative quality of each completion.

To optimize the policy, we first define the importance sampling ratio $\rho_t(\theta)$ between the current policy π_θ and the sampling policy $\pi_{\theta_{\text{old}}}$:

$$\rho_t(\theta) = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}. \tag{3}$$

Additionally, to ensure the policy remains close to the original behavior, we compute the token-level Kullback-Leibler (KL) divergence between the current policy π_θ and the reference policy π_{ref}:

$$\mathbb{D}_{\text{KL}}\left[\pi_\theta \| \pi_{\text{ref}}\right] = \pi_\theta(o_{i,t}|q, o_{i,<t}) \log \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\text{ref}}(o_{i,t}|q, o_{i,<t})}. \tag{4}$$

Alternatively, this can be approximated using the estimator $\frac{\pi_{\text{ref}}}{\pi_\theta} - \log \frac{\pi_{\text{ref}}}{\pi_\theta} - 1$ Schulman et al. (2017).

The overall GRPO objective is to maximize the expected group-relative advantage while penalizing deviations from the reference model. We define the total objective $\mathcal{J}_{\text{GRPO}}$ as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \hat{\mathbb{E}}_{i,t}\left[\mathcal{L}_t^{\text{CLIP}}(\theta) - \beta \mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}]\right] \tag{5}$$

where β is a coefficient controlling the strength of the KL regularization. To ensure training stability, we employ the clipped surrogate objective $\mathcal{L}_t^{\text{CLIP}}(\theta)$, defined as:

$$\min\left(\rho_t(\theta)\hat{A}_{i,t},\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\right) \tag{6}$$

Here, the clipping operator constrains the update magnitude relative to the old policy within the range $[1-\epsilon, 1+\epsilon]$, thereby preventing destructive policy updates.

B Latency & Real-Time Inference

A core strength of WiA-LLM is its ability to perform high-fidelity, interpretable counterfactual reasoning. However, as discussed in §3.4, LLM inference introduces significant latency, making it impractical to run on every game frame. To address this, we propose a two-pronged strategy: (1) a dual-system deployment architecture, and (2) knowledge distillation for scalable deployment.

B.1 Dual-System Deployment Architecture

We do not propose running the full WiA-LLM simulation for low-level, high-frequency actions (e.g., movement or skill targeting). Instead, the agent operates under a dual-system architecture that separates slow, deliberative reasoning from fast, reactive execution.

Dataset	URL
Ape210K Zhao et al. (2020)	https://github.com/Chenny0808/ape210k
MMLU Hendrycks et al. (2021)	https://huggingface.co/datasets/cais/mmlu
CEval Huang et al. (2023)	https://github.com/hkust-nlp/ceval
School-Chinese lanhin (2018)	https://github.com/lanhin/SchoolChinese
BBH Suzgun et al. (2023)	https://github.com/suzgunmirac/BIG-Bench-Hard
IfEval Zhou et al. (2023)	https://huggingface.co/datasets/google/IFEval
CharacterEval Tu et al. (2024)	https://github.com/morecry/charactereval

Table 3: Source links for the benchmarks

System 1 (Reactive Policy): A lightweight, model-free neural network handles real-time, low-latency control of the hero (e.g., movement and immediate tactical skill use). This policy can operate at the required environment frequency (e.g., ~ 30 ms per update).

System 2 (WiA-LLM Planner): WiA-LLM acts as the deliberative strategic layer. The planner is invoked only at key decision points that require long-horizon reasoning. Typical calls include: (i) **objective selection**, such as deciding whether to contest Dragon/Baron, push a high-ground tower, or rotate lanes; (ii) **rallying and grouping**, such as deciding whether to commit to a team fight; and (iii) **recall and resurrection planning**, such as choosing the hero's path and next objective after death or returning to base. This design reduces LLM inference frequency from every frame to roughly once every 5–10 seconds, or at major state transitions (e.g., an enemy hero death). In this way, the agent benefits from strategic planning without sacrificing real-time control.

B.2 Knowledge Distillation from WiA-LLM

The primary value of WiA-LLM is its ability to internalize complex game dynamics and predict strategic outcomes with high precision, though at a computational cost. To enable scalable deployment, we leverage knowledge distillation to transfer this strategic understanding into a much smaller, faster student model. Specifically, WiA-LLM (the teacher model) is used to generate a large, high-quality dataset of reasoning-augmented trajectories, including the current state, optimal high-level actions (from what-if analysis), and textual justifications. A much smaller, efficient neural network (the student model) is then trained to mimic the teacher's strategic decisions.

However, while we present this as the natural path to production deployment, the detailed implementation of knowledge distillation is beyond the scope of this work. Here, we focus on demonstrating the core capability of WiA-LLM to learn and execute explicit world modeling, leaving knowledge distillation for future work.

C WiA-LLM on General Benchmarks

To further verify that our models do not sacrifice their native language understanding and reasoning capabilities during training, we evaluate WiA-LLM on several standard benchmarks: **Ape210K** Zhao et al. (2020), **MMLU** Hendrycks et al. (2021), **CEval** Huang et al. (2023), **School-Chinese** lanhin (2018), **BBH** Suzgun et al. (2023), and **IfEval** Zhou et al. (2023).

C.1 Performance Analysis

Tables 4 and 5 show that our training approach generally preserves core language model capabilities, sometimes even enhancing logical reasoning and instruction-following. On math tasks (Ape210K), our 14B models remain stable (93.5 across all variants), while the 8B SFT-based models show only modest degradation. Subject exams (MMLU and CEval) demonstrate exceptional robustness, with performance variations within 1 percentage point, indicating factual knowledge is well preserved. Notably, logical reasoning (BBH) consistently improves with SFT-based training, with 8B models achieving 60.17–60.52 compared to the 58.35 baseline. In Table 5, we analyze instruction-following capabilities, finding that our method with GRPO enhances performance (8B: 0.379 vs. 0.351 on prompt-level accuracy), while SFT alone may cause slight degradation on the 8B model (0.301 vs. 0.351). These results confirm that our approach enables domain-specific improvements while

preserving essential language model abilities, with RL components proving particularly beneficial for maintaining instruction-following performance.

Model	Math	Memorization	Subject Exam		Logic
	Ape_210k	SchoolChinese	MMLU	CEval	BBH
qwen3-14b	**93.5**	91.85	80.21	82.76	**65.48**
WiA-LLM (14b-sft)	92.0	**92.28**	80.18	83.06	**65.48**
WiA-LLM (14b-grpo)	**93.5**	92.19	**80.56**	82.91	65.13
WiA-LLM (14b-sft-grpo)	**93.5**	91.69	80.25	**83.14**	65.30
qwen3-8b	93.0	88.0	75.96	78.08	58.35
WiA-LLM (8b-sft)	90.5	87.02	76.60	**78.68**	**60.52**
WiA-LLM (8b-grpo)	**93.5**	**88.04**	76.04	78.45	57.83
WiA-LLM (8b-sft-grpo)	89.5	86.72	**76.631**	78.6	60.17

Table 4: Performance of different models on math, academic, general knowledge, and logical reasoning benchmarks.

Model	Prompt-level loose-acc	Inst-level loose-acc	Prompt-level strict-acc	Inst-level strict-acc
qwen3-14b	0.357	0.494	0.338	0.475
WiA-LLM (14b-grpo)	0.355	0.496	0.342	0.482
WiA-LLM (14b-sft)	0.355	0.498	0.336	0.480
WiA-LLM (14b-sft-grpo)	0.362	0.501	0.344	0.486
qwen3-8b	0.351	0.495	0.340	0.483
WiA-LLM (8b-grpo)	0.379	0.514	0.366	0.500
WiA-LLM (8b-sft)	0.301	0.440	0.290	0.429
WiA-LLM (8b-sft-grpo)	0.336	0.477	0.323	0.461

Table 5: Performance on the IFEval benchmark Zhou et al. (2023). Detailed metrics are provided in Appendix C.3.

C.2 Benchmark Descriptions

The details of the benchmarks are as follows. To facilitate reproducibility, we provide the source links for these benchmarks in Table 3.

- **Ape210K** Zhao et al. (2020): A large-scale, template-rich math word problem dataset. For our experiments, we randomly sample 200 examples from the test set.
- **MMLU** Hendrycks et al. (2021): A comprehensive benchmark covering knowledge from 57 subjects across STEM, humanities, social sciences, and more. It ranges in difficulty from elementary to advanced professional level, testing both world knowledge and problem-solving ability. We sample the first 50 examples from each subject, resulting in $50 \times 57 = 2850$ cases for our experiments.
- **CEval** Huang et al. (2023): Similar to MMLU, CEval is a Chinese-language benchmark comprising 52 subtasks across four categories: STEM, social sciences, humanities, and others. We use it as an additional testbed to evaluate language mixing challenges as discussed in DeepSeek-AI et al. (2025).
- **School-Chinese** lanhin (2018): This benchmark assesses the memorization capabilities of LLMs on classical Chinese poetry by requiring the model to predict subsequent content given introductory text. We manually collected these datasets from public repositories, resulting in a benchmark with 269 samples.
- **BBH** Suzgun et al. (2023): A subset of BIG-Bench Srivastava et al. (2023) focused on 23 challenging tasks that require multi-step reasoning. It is widely regarded as a standard evaluation set for assessing the logical reasoning abilities of language models.

- **IfEval** Zhou et al. (2023): A standard benchmark for evaluating the instruction-following capabilities of LLMs. It contains approximately 500 verifiable instructions, such as "write more than 400 words" or "mention the keyword 'AI' at least three times," which can be automatically checked using heuristics.

C.3 METRICS FOR IFEVAL

We use the IFEval benchmark Zhou et al. (2023) to evaluate the instruction-following capabilities of WiA-LLM, as reported in Table 5. The evaluation is based on four key metrics, which vary along two dimensions: **granularity** (whether evaluation is performed at the level of the entire prompt or on individual instructions) and **verification strictness** (how rigorously the model's output is checked).

For granularity, we consider two levels: (1) *Prompt-Level*: The entire prompt (which may contain multiple instructions) is evaluated as a single unit. All instructions must be satisfied for the prompt to be counted as correct. (2) *Instruction-Level*: Each instruction is evaluated independently, regardless of the prompt to which it belongs.

For verification strictness, we also consider two levels: (1) *Strict*: The output must match the instructions exactly, including both content and formatting. Even minor deviations (e.g., missing bold text) result in failure. (2) *Loose*: Some flexibility is allowed by applying transformations to the output (e.g., removing markdown or extra text) prior to evaluation, focusing on the main intent of the instruction.

The precise definitions of each metric are as follows:

- **Prompt-Level Strict-Accuracy**: The percentage of prompts for which every instruction is followed exactly as specified, with no deviations in content or formatting. For prompts with multiple instructions, all must be perfectly executed for the prompt to be counted as correct; a single error causes the prompt to fail.
- **Instruction-Level Strict-Accuracy**: The percentage of individual instructions across all prompts that are followed exactly as specified. Each instruction is checked independently, and the metric counts how many are perfectly executed, even if others in the same prompt fail.
- **Prompt-Level Loose-Accuracy**: A more lenient version of prompt-level accuracy. After normalizing the output (e.g., removing markdown or extraneous text), all instructions must be satisfied for the entire prompt to be counted as correct.
- **Instruction-Level Loose-Accuracy**: A more relaxed version of instruction-level accuracy. After normalization, each instruction is evaluated separately and considered correct if it meets the requirements, even if there are minor formatting differences.

D PROMPTING LIST

Training Prompt. To train WiA-LLM, we begin by designing a straightforward template that guides the initial LLM to follow our predefined instructions. As shown in Table D, this template organizes the model's output into two parts: a reasoning process and a final answer. We intentionally restrict our requirements to this structural format, following Jin et al. (2025), in order to avoid introducing any content-specific biases.

Training Prompt

```
You are an AI assistant for the Honor of Kings game. As the
 ↪  main player's assistant, you need to analyze the
 ↪  potential battlefield changes #time_gap# seconds after
 ↪  the main player's action.

The current game state is:
 ↪  <game_state>GAME_STATE</game_state>, and the main
 ↪  player's executed action is <action>ACTION</action>.

Please consider the following four aspects when analyzing the
 ↪  potential game state change:
  - 1. Minion Wave Changes: Analyze changes in minion wave
   ↪  pushing status (e.g., whether a lane's minions enter or
   ↪  exit enemy turret range) and changes in wave-clearing
   ↪  heroes.
  - 2. Turret Changes: Analyze changes in turret HP and
   ↪  changes in turret attack status. Note that turret
   ↪  protection mechanics do not grant invincibility;
   ↪  turrets still lose HP when attacked.
  - 3. Hero Changes: Analyze hero deaths (e.g., a hero being
   ↪  eliminated).
  - 4. Dragon Status Changes: Analyze whether the Lord,
   ↪  Turtle, or Storm Dragon is being attacked. Note that
   ↪  any HP reduction is considered an attack.

Please put your thinking process in <think></think>, and game
 ↪  state change in <answer></answer>. The answer should be
 ↪  formulated as a JSON object covering the following keys:
 ↪  (1) minion_wave_changes; (2) turret_changes; (3)
 ↪  hero_changes and (4) dragon_status_changes.
```

Distillation Prompt from DS-R1. In Table D, we present the prompt used to distill reasoning data from Deepseek-R1. We supply the ground-truth game state change along with the corresponding instructions to the R1 model, prompting it to generate the reasoning process (C_t) that links the state (S_t) and action (a_t) to the resulting outcome.

Distillation Prompt from DS-R1

```
You are an AI assistant for the Honor of Kings game. Your
 ↪  task is to analyze the given battlefield changes and
 ↪  generate the corresponding logical reasoning process
 ↪  explaining how these changes occurred based on the
 ↪  current game state.

Current game state: <game_state>GAME_STATE</game_state>; main
 ↪  player's executed action: <action>ACTION</action>;
```

```
Game state changes after #time_gap#:
 ↪  <game_state_change>STATE_CHANGE</game_state_change>.

As a game assistant, you need to analyze the causes of the
 ↪  battlefield changes that occurred #time_gap# seconds
 ↪  after the main player's action and generate a reasoning
 ↪  process explaining how these changes happened based on
 ↪  the current game state.

Place your reasoning inside <answer></answer>.
```

Downstream Task Prompt. In Table D, we present the prompt used for the downstream task of action prediction. The model first selects the top four potential actions and then analyzes the consequences of each to determine the optimal choice.

Downstream Task Prompt

```
Given real-time game state information from an MOBA game,
 ↪  provide decision-making suggestions as an assistant to
 ↪  the main player.

# Board State: <game_state>GAME_STATE</game_state>

Please first select the 4 most probable actions from the
 ↪  candidate options set
 ↪  <action_candidates>ACTION_CANDIDATES</action_candidates>.

Then analyze the consequences of each action one by one.

Finally, select the action most beneficial to our team based
 ↪  on whether the consequences are favorable to our
 ↪  situation.

Please put your thinking process in <think></think>, and
 ↪  actions in <answer></answer>.
```

Action Space for Downstream Tasks

Category	Action	Explanation
None	None	No action triggered for a short period
Dragon	Lord	Deal damage to the Lord (Main Dragon)
	Tyrant	Deal damage to the Tyrant (Early Game Dragon)
	Dragon King	Deal damage to the Dragon King (Late Game Dragon)
Tower	Crystal	Deal damage to enemy Crystal (Nexus)
	Top Tower	Deal damage to Top Lane Tower
	Mid Tower	Deal damage to Mid Lane Tower
	Bot Tower	Deal damage to Bottom Lane Tower
Defense	Defend Crystal	Defend our Crystal
	Defend Top Tower	Defend Top Lane Tower
	Defend Mid Tower	Defend Mid Lane Tower
	Defend Bot Tower	Defend Bottom Lane Tower
Hero	Top/Mid/Bot Hero	Damage enemy heroes in respective lanes
	River Top/Bot Hero	Damage enemies in River areas
	Allied/Enemy Jungle Hero	Damage enemies in Jungle areas
	Ally High-ground Hero	Damage enemies on our High-ground
	Enemy High-ground Hero	Damage enemies on enemy High-ground
Line	Top/Mid/Bot Minions	Clear minions in respective lanes
	Ally High-ground Minions	Clear minions on our High-ground
	Enemy High-ground Minions	Clear minions on enemy High-ground
Buff	Allied Red/Blue	Take our Red/Blue Buff
	Enemy Red/Blue	Steal enemy Red/Blue Buff
Jungle	Allied/Enemy Camps	Clear non-buff camps
	Void Spirit / Crimson Raptor	Kill River objectives (Crabs)
Grouping	Lane Grouping	Group in Top, Mid, or Bot Lane
	River Grouping	Group in Upper or Lower River
	Jungle Grouping	Group in Allied or Enemy Jungle
	High-ground Grouping	Group on Allied or Enemy High-ground
Recall	Recall	Hero at fountain (including walk-back)