Guidelines as Environments: A World Model Approach to Rule Following

Haiqing Li, Wenliang Zhong, Yinhao Wu, Hehuan Ma, Yuzhi Guo, Thao M. Dang, Junzhou Huang*
University of Texas at Arlington, Arlington, Texas, USA
{hxl9110,wxz9204,yxw2120,hehuan.ma,yuzhi.guo,tmd4090}@mavs.uta.edu
{jzhuang}@exchange.uta.edu

Abstract

Guideline-following is increasingly important in compliance, customer support, and other regulated workflows, where correctness is defined by explicit rule systems rather than heuristics. Learning to follow guidelines is challenging because guidelines are interdependent: rules can trigger, suppress, or conflict with one another, while locally plausible responses may violate global constraints. Most existing methods treat guidelines as static text and rely on implicit reasoning or deeper decoding, making rule interactions and satisfaction status hard to observe and control. A more feasible approach is to model guideline execution with an explicit state that tracks evolving rule evidence across steps. However, conventional world models are a poor fit: they typically assume privileged feedback or well-defined transition dynamics, assumptions that do not hold when reasoning occurs purely in language space under ambiguous, text-defined constraints. As a solution, we propose RGCWM, a **R**ule-**G**rounded **C**ausal **W**orld **M**odel that builds an explicit state space from the guideline text itself. RGCWM represents rule applicability and satisfaction as a continuously updated evidence state, externalizes inter-rule dependencies as a causal structure, and plans at inference time by counterfactually evaluating candidate responses under model-estimated state transitions. Experiments show that this shift from implicit text reasoning to state-based reasoning enables stable, controllable execution of complex interacting rules across diverse domains.

1 Introduction

The instruction-following capability of Large Language Models (LLMs) (Achiam et al., 2023; Guo et al., 2025; Yang et al., 2025) has enabled them to perform diverse tasks by accepting different prompts without specific training (Sun et al., 2025; Kuang et al., 2025). However, in real-world scenarios, some domain-specific tasks do not require following a simple instruction, but a complex procedure of rules (Diao et al., 2025; Jiang et al., 2024). For example, consider applying different discounts while shopping: Rule A grants a discount for purchases over $100, while Rule B excludes clearance items from the threshold. For a $120 cart with $50 clearance, Rule B blocks Rule A, and no discount should apply. Yet even nowadays proprietary LLMs often grant the discount by matching the surface total "$120" to Rule A, failing to account for this blocking interaction.

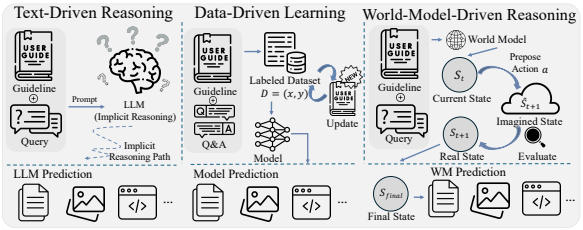

Figure 1: **Text-driven reasoning** uses guidelines as static context and relies on implicit LLM reasoning. **Data-driven learning** internalizes guidelines via annotated data and parameter updates. **World-model-driven reasoning** externalizes guidelines into an explicit world model, maintains rule states, and simulates state transitions to enforce consistent adherence.

Without loss of generality, we term different procedures of following domain-specific rules as guidelines in this paper. Most guidelines are not static: they update frequently, encode prerequisites and conflicts, and impose logical constraints that are intolerant to violations (Wen et al., 2025). Unfortunately, most LLM successes stem from data-driven learning on large-scale corpora; this paradigm can be misaligned with correctness defined by explicit rules (Bai et al., 2022; Dong et al., 2024), especially when guidelines contradict an LLM's internal priors and commonsense heuristics, yielding fluent outputs that violate the guideline system's implicit logic. Recent research trying

*Corresponding authors

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16302–16318

to address guideline-following largely inherits the paradigm of instruction following, framing guidelines as textual conditions to be reasoned about during generation. Chain-of-Thought (CoT) (Diao et al., 2025) and Self-Consistency (Prasad et al., 2024) unfold, or aggregate intermediate reasoning traces to stabilize outputs. Textual guideline-based approaches such as GoLLIE (Sainz et al., 2023) similarly incorporate annotation guidelines directly into the prompt as natural-language constraints. While single LLMs may not be enough for guideline-following, some research explores LLM agents. Tool-augmented agents (Cheng et al., 2024) and ReAct-style frameworks (Yao et al., 2022) further interleave reasoning with action execution, enabling interaction with external tools or environments, while Self-Reflection (Shinn et al., 2023) extends this approach through critique or iterative revision in natural language space. Across these approaches, guideline adherence is treated as an emergent property of reasoning and generation, rather than the outcome of an explicit decision process over interacting rules; rule validity and interaction effects are therefore left unmodeled, a failure mode we characterize as causal blindness.

Rather than treating guidelines as static textual context, we argue that they should be modeled as an explicit environment. Guideline-following then becomes a problem of stateful planning (Huang et al., 2022): the agent maintains a rule state and evaluates candidate responses by simulating their downstream effects on global rule satisfaction, rather than by merely extending a text sequence. This formulation aligns guideline reasoning with world-model-based decision making (Hafner et al., 2020), where actions induce transitions in a structured state space. However, naive simulation or sequential replanning can incur prohibitive inference-time cost, while compliance signals inferred from language are inherently noisy. A viable solution shall therefore support efficient counterfactual evaluation, enforce global rule consistency, and remain robust to uncertainty, while avoiding any reliance on additional training.

To address these challenges, we propose **RGCWM**, a **R**ule-**G**rounded **C**ausal **W**orld **M**odel for reliable guideline following. We first externalize guidelines into a static Rule Causal Graph that encodes typed dependencies, making cross-rule interactions explicit. However, structure alone is insufficient at inference time: an agent shall also track which rules are currently applicable and satisfied under a partial trajectory. RGCWM therefore maintains a dynamic WorldState with a compact history abstraction and a continuous rule-evidence vector. It performs inference-time planning by evaluating candidate responses via a *1-Call Rollout* that predicts their downstream effects from the same state in a single batched inference. Moreover, global planning can still leave residual local issues (e.g., minor violations or phrasing-level inconsistencies); to repair these without re-running rollout, we optionally apply a lightweight Proposer-Critic-Refiner (PCR) loop (Lightman et al., 2023) for localized edits under the fixed graph. We evaluate RGCWM across multiple backbones and task categories. RGCWM consistently outperforms state-of-the-art baselines in a wide range of categories.

Overall, this work is the first to (1) identify causal blindness as a fundamental limitation of text-based guideline reasoning; (2) reformulate guideline-following as stateful planning over an explicit causal rule system; and (3) introduce RGCWM, a world-model-driven framework that achieves robust, backbone-agnostic gains without additional training.

2 Related Work

Inference-Time Reasoning. CoT (Wei et al., 2022) elicits explicit intermediate steps, while self-consistency (Prasad et al., 2024) aggregates multiple sampled traces to improve robustness. Guideline Forest (Chen et al., 2025) further induces reusable high-level guidelines from verified trajectories and performs multi-guideline execution with stepwise aggregation at inference time. ReAct (Yao et al., 2022) interleaves reasoning with action execution, enabling tool use and multi-step interaction. Across these approaches (Yao et al., 2023), adherence is governed by reasoning traces and action sequences, without an explicit rule state that tracks rule validity and inter-rule dependencies over steps.

Textual and Retrieval-Based Methods. Textual and retrieval-based methods incorporate guidelines as input context, where generation is conditioned on the provided rules. GoLLIE (Sainz et al., 2023) incorporates annotation guidelines directly into the prompt as natural-language constraints, enabling zero-shot extraction. Retrieval-based agents (Pang et al., 2023; Dong et al., 2024; Xu et al., 2023) improve rule coverage by selecting relevant rules on demand; however, the retrieved rules function

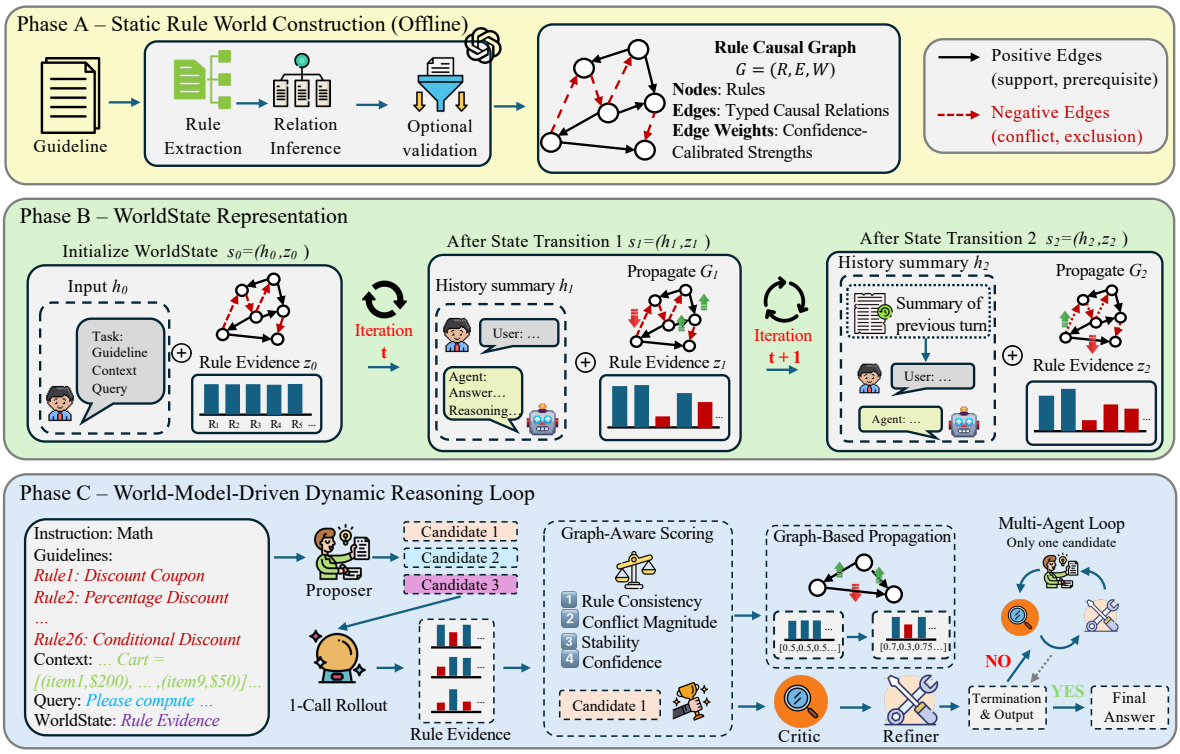

Figure 2: Overview of RGCWM as a causal world model for guideline following. Guidelines are externalized into a static Rule Causal Graph (Phase A), tracked as a dynamic WorldState with explicit state transitions (Phase B), and used for model-based planning via rollout, consistency-aware scoring, and propagation (Phase C). This design enables counterfactual evaluation of actions and stable reasoning under interdependent and conflicting rules.

as contextual snippets rather than as a stateful, dependency-coupled constraint representation.

State, Transitions, and Causality. World-model approaches in model-based reinforcement learning (RL) and symbolic MDPs (Hafner et al., 2020) enable planning via explicit states and transitions, but are typically defined over physical or task environments (Hao et al., 2023). Recent work links LLMs with causal world models (Kiciman et al., 2023; Ban et al., 2025), but usually operates over learned representations rather than explicit rule states. Meanwhile, neural causal and planning work often learns or assumes causal structure and noisy relational dynamics (Goodman et al., 2007; Guerdan et al., 2023; Lang and Toussaint, 2010; Wang et al., 2018); in contrast, guideline-following relies on text-specified rule relations and thus requires tracking a rule-compliance state under language actions.

3 Guideline-Following Paradigms

We consider a guideline-following task T with input space $\mathcal{X}$, output space $\mathcal{Y}$, and guideline S, typically a collection of rules. We model S as a set of atomic rules $\mathcal{R} = \{r_1, \ldots, r_{|\mathcal{R}|}\}$. Given $x_q \in \mathcal{X}$, the model outputs $y_q \in \mathcal{Y}$ that solves the task and complies with S.

Text-Driven Reasoning (Fig.1 left). In this paradigm, the guideline S is appended as context to a LLM L_θ through a prompt template P, thereby producing the output $\hat{y}_q = L_\theta(P(S, x_q))$. In multi-turn reasoning scenarios, the model context is composed purely of text: the initial context is $c_0 = x_q$, the output at step t is $\hat{y}_t = L_\theta(P(S, c_t))$, and this output is concatenated back into the context to form $c_{t+1} = c_t \oplus \hat{y}_t$. Under this paradigm, no explicit rule state is maintained; rule compliance and cross-rule interactions rely implicitly on the internal representations of the model L_θ.

Data-Driven Learning (Fig. 1 middle). In the data-driven paradigm, guideline compliance is learned offline from data rather than enforced at inference. Annotators construct a supervised set $\mathcal{D}_M = \{(x_i, y_i)\}_{i=1}^M \subset \mathcal{X} \times \mathcal{Y}$ with y_i compliant with S, and train f_ϕ by minimizing the empirical risk: $\phi^\star = \arg\min_\phi \frac{1}{M} \sum_{i=1}^M \ell(f_\phi(x_i), y_i)$, yielding $\hat{y}_q = f_{\phi^\star}(x_q)$. Classical RL is a sequential variant that learns a policy from transitions $\mathcal{D}_{\text{RL}} = \{(s_t, a_t, r_t, s_{t+1})\}$ via return maximization $\psi^\star = \arg\max_\psi \mathbb{E}_{\pi_\psi}\left[\sum_{t\geq 0} \gamma^t r_t\right]$. In both cases, rule effects are absorbed into parameters, so guideline updates or long-horizon conditions

typically require new data and retraining.

World-Model-Driven Reasoning (Fig. 1 right). In this paradigm, the guideline S is treated as an explicit environment rather than static text or implicit training signal. Rules are externalized into a structured system with explicit dependencies (e.g., support, prerequisite, conflict, exclusion), and the agent reasons over a state that represents the current configuration of rule applicability and compliance. Decision making is formulated as stateful planning: candidate actions are evaluated by simulating their potential effects on the guideline state, and updates reflect how enforcing one rule may activate, suppress, or invalidate others. By making rule interactions and compliance status explicit, this paradigm enables reasoning with interdependent and conflicting guidelines through direct state evolution, rather than relying on implicit textual inference or data absorption.

4 Problem Formulation

Given a query x_q and a guideline specification S, we model guideline following as stateful planning over a rule system. In practice, guidelines are often itemized for interpretability; thus we abstract S (a rule list or a text document) into atomic rules $\mathcal{R} = \{r_i\}_{i=1}^{|\mathcal{R}|}$ and a fixed dependency graph $\mathcal{G} = (\mathcal{R}, \mathcal{E})$. At step t, the agent maintains a world state $s_t = (h_t, z_t)$, where h_t is a compact history summary and $z_t \in [0,1]^{|\mathcal{R}|}$ stores per-rule satisfaction evidence. An action a_t is a language-level candidate response chosen from a finite set $\mathcal{A}_t$. The state evolves via a deterministic transition

$$s_{t+1} = F(s_t, a_t; \mathcal{G}, S), \tag{1}$$

until termination, and the final output is the last selected action. $F(\cdot)$ denotes the abstract state transition induced by a reasoning action under the fixed rule system (instantiated by our inference procedure in Sec. 5).

5 The RGCWM Framework

Static Rule Causal Graph Construction. We formalize the guideline input S as a set of explicit, itemized rules $\mathcal{R} = \{r_i\}_{i=1}^{|\mathcal{R}|}$ and infer a latent dependency structure over these atomic units. *Relation Inference:* Rule dependencies are modeled as a directed graph $\mathcal{G}(\mathcal{R}, \mathcal{E}, \mathcal{W})$. For each rule pair (r_i, r_j), a frozen LLM predicts a relation label τ_{ij} and a confidence score c_{ij}, aggregated over three independent queries (Li et al., 2024). We define positive relations $\mathcal{T}^+ = \{\texttt{support}, \texttt{prerequisite}\}$ and negative relations $\mathcal{T}^- = \{\texttt{conflict}, \texttt{exclusion}\}$. An edge e_{ij} is retained only if (i) at least two of the three predictions agree on the same non-none label, and (ii) the aggregated confidence $\bar{c}_{ij}$ is at least 0.7.

Algorithm 1 RGCWM Inference (Simplified. Full Version in Appendix B)

Require: Guideline S, fixed graph $\mathcal{G}$, query x_q, frozen LLM L_θ

Initialize h_0, z_0; set $\tilde{a}_{-1} \leftarrow \emptyset$
for $t = 0, 1, \ldots, T_{\max} - 1$ **do**
  if $t = 0$ **then**
    $\mathcal{A}_0 = \{a_0^{(k)}\}_{k=1}^{K} \leftarrow \textsc{Proposer}(x_q, h_0, z_0, S; L_\theta)$
    $(k^\star, \hat{z}_1) \leftarrow \textsc{RolloutSelect}(x_q, h_0, z_0, \mathcal{A}_0, S; \mathcal{G}, L_\theta)$
    $\tilde{a}_0 \leftarrow a_0^{(k^\star)}$
  else
    $\tilde{a}_t \leftarrow \textsc{ProposerSingle}(x_q, h_t, z_t, \tilde{a}_{t-1}, S; L_\theta)$
    $\hat{z}_{t+1} \leftarrow z_t$ (no rollout; default local effect)
  end if
  if UsePCR **then**
    $(\tilde{a}_t, \hat{z}_{t+1}) \leftarrow \text{PCR}(x_q, h_t, \tilde{a}_t, z_t, S; L_\theta)$
  end if
  $z_{t+1} \leftarrow \textsc{Propagate}(z_t, \hat{z}_{t+1}; \mathcal{G})$
  $h_{t+1} \leftarrow \textsc{Summarize}(h_t, \tilde{a}_t)$
  if $\textsc{Terminate}(t, z_{t+1}, \tilde{a}_t)$ **then**
    return $\tilde{a}_t$
  end if
  $h_t \leftarrow h_{t+1}$; $z_t \leftarrow z_{t+1}$; $\tilde{a}_{t-1} \leftarrow \tilde{a}_t$
end for
return $\tilde{a}_{T_{\max}-1}$

Each retained edge is assigned a signed weight

$$w_{ij} = \begin{cases} g(\bar{c}_{ij}), & \tau_{ij} \in \mathcal{T}^+ \\ -g(\bar{c}_{ij}), & \tau_{ij} \in \mathcal{T}^- \end{cases} \tag{2}$$

where the sign encodes supportive versus conflicting relations, $g(\cdot)$ is an identity mapping, i.e., $g(u) = u$. The resulting graph topology remains fixed during inference and serves as an immutable structural prior. We do not assume the inferred causal graph or rule-evidence states to be ground-truth or perfectly calibrated; both are treated as approximate latent structures induced by the guideline specification and estimated by frozen LLMs. We analyze reliability, robustness, and reproducibility under such estimation noise in Appendix A (graph perturbations in Appendix A.1, evidence perturbations in Appendix A.2, and determinism settings in Appendix A.6).

Causal Planning via 1-Call Rollout. As illustrated in Fig. 2, each reasoning step operates on a minimal world state $s_t = (h_t, z_t)$, where h_t summarizes the reasoning history and $z_t \in$

$[0,1]^{|\mathcal{R}|}$ encodes rule-level evidence, estimated by the same frozen LLM used in the 1-Call Rollout (Appendix A.3), under a fixed guideline S and graph $\mathcal{G}$. We interpret $z_t(i)$ as evidence that rule r_i is currently applicable and satisfied under the trajectory; correspondingly, low values indicate either likely violation or inapplicability rather than a pure probability. At step t, a Proposer conditioned on (h_t, z_t, S) generates a small candidate set $\mathcal{A}_t = \{a_t^{(k)}\}_{k=1}^{K}$. Rather than committing to a single action, RGCWM performs world-model-based planning by querying a frozen LLM to estimate, for each candidate $a_t^{(k)}$, a hypothetical future rule-evidence state $\hat{z}_{t+1}^{(k)}$ capturing potential rule activations, suppressions, or violations. All candidates share the same state (h_t, z_t) at rollout time and are evaluated jointly in a single batched inference, enabling efficient counterfactual screening.

Each predicted state $\hat{z}_{t+1}^{(k)}$ is scored by a graph-aware structural validity function that aggregates four complementary signals:

$$\begin{aligned}\text{Score}_k \triangleq \lambda_1 C_k + \lambda_2 \exp\big(-\gamma\,\text{Conf}_k\big) \\ + \lambda_3 \text{S}_k + \lambda_4 L_k.\end{aligned} \tag{3}$$

Here, rule consistency C_k measures overall rule satisfaction, defined as the average of per-rule evidences:

$$C_k = \frac{1}{|\mathcal{R}|}\sum_{i=1}^{|\mathcal{R}|} \hat{z}_{t+1}^{(k)}(i). \tag{4}$$

Aggregated conflict magnitude penalizes co-activation of incompatible rules:

$$\text{Conf}_k = \sum_{(i,j)\in\mathcal{E}^-} |w_{ij}|\,\hat{z}_{t+1}^{(k)}(i)\hat{z}_{t+1}^{(k)}(j). \tag{5}$$

Stability measures self-consistency under dependencies encoded in $\mathcal{G}$:

$$\begin{aligned}\text{S}_k = \frac{1}{|\mathcal{R}|}\sum_j \Big(1 - \Big|\hat{z}_{t+1}^{(k)}(j) \\ - \sigma\Big(\sum_i w_{ij}\hat{z}_{t+1}^{(k)}(i)\Big)\Big|\Big),\end{aligned} \tag{6}$$

where $\mathcal{E}^-$ denotes negative edges; $\sigma(\cdot)$ is the sigmoid, acting as a heuristic consistency check. Finally, $L_k \in [0,1]$ is a lightweight, model-reported confidence cue from the same rollout, used with a small weight and not treated as calibrated uncertainty (Appendix A.4).

The planner selects the action with the highest structural validity,

$$a_t^\star = \arg\max_{a_t^{(k)}\in\mathcal{A}_t} \text{Score}_k, \tag{7}$$

and outputs the corresponding $\hat{z}_{t+1}^{(\star)}$ for subsequent graph-based state update. Intuitively, 1-Call Rollout evaluates all candidates from the same world state under a fixed causal structure, avoiding trajectory drift from sequential re-planning; the joint graph-aware scoring enables global counterfactual comparison rather than incremental correction, yielding more stable decisions under interdependent rules.

The structural validity score is a heuristic aggregation designed for inference-time planning rather than a theoretically optimal objective; its components capture complementary aspects of rule compliance and are empirically validated via term ablations (Appendix A.5).

Graph Propagation for Dynamic Rule Update. The local estimate $\hat{z}_{t+1}^{(\star)}$ captures the predicted effect of the selected action in isolation and does not account for inter-rule dependencies. To incorporate higher-order interactions encoded in $\mathcal{G}$, we propagate state changes rather than absolute evidence over the fixed causal structure.

Let $\Delta z_t = \hat{z}_{t+1}^{(\star)} - z_t$ denote the action-induced change in rule evidence. The next state is computed via a deterministic propagation operator:

$$z_{t+1}(j) = \text{clip}\left(\hat{z}_{t+1}^{(\star)}(j) + \beta\sum_i w_{ij}\,\Delta z_t(i)\right), \tag{8}$$

where $\text{clip}(u) = \min(1, \max(0, u))$. Propagating Δz_t avoids repeatedly amplifying already-satisfied rules and yields stable updates in practice. We use a step size $\beta \in (0,1]$ and apply element-wise clipping to keep $z_{t+1} \in [0,1]$. This operation updates only the dynamic rule-evidence state and never modifies the graph topology. Graph propagation is applied immediately after candidate selection, and re-applied after PCR if the Refiner alters the action, using the re-evaluated local effects $\hat{z}_t$. We report hyperparameter sensitivity analyses for β in Appendix A.8 (Table 12).

PCR Refinement. Following rollout selection $(a_t^\star)$, RGCWM executes an iterative PCR loop (Madaan et al., 2023) under the fixed graph $\mathcal{G}$. At each step, the **Proposer** generates a single action conditioned on the current state $s_t = (h_t, z_t)$.

The **Critic** audits this proposal for residual violations or logical gaps, producing structured feedback. The **Refiner** then applies minimal edits to yield $\tilde{a}_t$, or defaults to a no-op if confidence is high. Crucially, PCR avoids re-planning (i.e., no candidate resampling or rollout re-evaluation). Instead, PCR returns the refined action together with an updated local evidence estimate, $(\tilde{a}_t, \hat{z}_t) \leftarrow \mathrm{PCR}(x_q, h_t, a_t^\star, z_t, S; L_\theta)$, followed by a state update via Eq. (8). The updated state (h_{t+1}, z_{t+1}) then conditions the next Proposer step, forming a state-driven single-action loop. Throughout PCR, $\mathcal{G}$ remains immutable while only s_t evolves.

Termination and Convergence-Driven Answer Finalization. RGCWM does not employ a separate answer selection module. Instead, the final answer emerges when the latent world state converges. At each iteration t, the system monitors explicit termination criteria over the evolving world state $s_t = (h_t, z_t)$. These conditions are checked in a prioritized order, with convergence-based criteria evaluated before the iteration budget.

Specifically, the reasoning process terminates if any of the following holds:

1. **Rule consistency saturation.** Terminate when $\mathrm{C}(z_{t+1}) \geq \tau_c$, indicating that the updated state sufficiently satisfies the guideline set, where $\mathrm{C}(z) = \frac{1}{|\mathcal{R}|}\sum_i z(i)$.
2. **Marginal state improvement.** Terminate when $\left|\mathrm{Score}_t^{\mathrm{struct}} - \mathrm{Score}_{t-1}^{\mathrm{struct}}\right| \leq \tau_\Delta$, suggesting further iterations are unlikely to yield meaningful gains.
3. **Action stability.** Terminate when the finalized action remains unchanged for K consecutive iterations, i.e., $\tilde{a}_t = \tilde{a}_{t-1} = \cdots = \tilde{a}_{t-K+1}$.
4. **Iteration budget.** Terminate when the maximum number of iterations $T_{\max}$ is reached to ensure bounded computational cost.

Once termination is triggered, the system returns the current action and its associated reasoning trajectory as the final output, without any additional ranking or selection across iterations. This ensures that answers arise as a direct consequence of world-state stabilization, rather than post hoc selection among competing candidates.

6 Experiments

Benchmarks. We evaluate RGCWM on GUIDEBENCH (Diao et al., 2025), a benchmark for domain-oriented guideline following. GuideBench spans seven task categories, including *audit algorithm* (dataset not publicly released), *price matching*, *text relevance*, *math*, *agent chatting*, *summarization*, and *hallucination detection*. Each instance is formulated as either a question–answer or multiple-choice task, consisting of an *Instruction*, a set of *Guidelines*, and a *Context* passage; multiple-choice tasks additionally provide candidate options.

In the original guideline setup, each instance is accompanied only by a curated subset of rules. In realistic deployments, however, an agent shall retrieve or reason over the entire guideline to identify the applicable rules, which is substantially more challenging: the model shall operate in a dense logical network formed by the interplay of all rules. Unlike the original setting, we adopt a full-rule setting in which each instance is provided with the complete guideline during inference. This setting requires the model to dynamically infer which rules are applicable from the full rule space while maintaining global consistency under complex prerequisites, conflicts, and exclusions. At the same time, it enables a systematic evaluation of robustness and scalability under conditions that more closely reflect real-world deployment.

Baselines. We compare RGCWM against representative inference-time baselines spanning: (i) standard prompting (direct answering), (ii) reasoning prompting (CoT, CoA (Pan et al., 2024)), (iii) self-improvement via self-critique/revision (Reflexion, ReAct), (iv) multi-agent deliberation (Debate (Hu et al., 2025), Collaboration (Zhang et al., 2025)), (v) verbal RL-style methods leveraging interaction histories or preference signals (Memento (Zhou et al., 2025), Training-Free-GRPO (Cai et al., 2025)), and (vi) test-time compute scaling under a fixed model (Compute-Optimal (Snell et al., 2025)). All methods are evaluated with identical inputs, guideline settings, and decoding configurations.

Implementation Details. All experiments are inference-time only with frozen LLMs. We fix the structural-score weights to $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.4, 0.3, 0.2, 0.1)$ and set $\gamma = 1$ throughout. We evaluate two backbones, **Qwen3-8B** (Yang et al., 2025) and **DeepSeek-R1** (Guo et al., 2025). The rule-evidence vector is initialized as $z_0 = 0.5$. We use $K = 3$ candidates and $T_{\max} = 3$ iterations. Early stopping follows Sec. 5 and triggers when any criterion holds: $\mathrm{C}(z_{t+1}) \geq 0.95$, $|\Delta\mathrm{Score}_t^{\mathrm{struct}}| \leq \tau_\Delta$ with $\tau_\Delta = 0.01$, or the se-

Method	Qwen3-8B							DeepSeek-R1						
	All	Price	Text	Math	Chat	Sum.	Halluc.	All	Price	Text	Math	Chat	Sum.	Halluc.
Standard Prompt	73.3	68.1	74.5	15.4	95.0	55.2	93.2	78.2	72.6	90.6	15.4	93.9	62.1	90.7
Reasoning Prompting														
CoT	73.5	68.1	75.0	15.4	95.6	55.2	93.2	78.3	72.6	90.6	15.4	93.9	62.1	91.5
CoA (ICLR'25)	78.0	74.2	78.6	26.9	95.0	63.8	94.9	85.1	81.5	88.5	57.5	96.7	75.9	92.4
GuideBench (ACL'25)	77.5	71.5	84.9	25.0	94.4	62.1	92.4	84.9	80.3	89.6	69.2	94.4	72.4	93.2
Self-Improvement Prompting														
Reflexion (NeurIPS'23)	75.9	69.0	77.6	30.8	96.7	60.3	94.9	82.3	79.2	83.3	67.3	95.0	67.2	86.4
ReAct (ICLR'23)	76.3	73.3	78.1	**36.5**	92.8	48.3	90.7	83.4	82.6	85.9	69.2	93.9	60.3	83.9
Multi-Agent														
Debate (NeurIPS'25)	79.6	75.6	82.3	26.9	96.7	69.0	92.4	86.8	84.2	88.5	59.6	98.9	75.9	96.6
Collaboration (ACL'25)	77.6	72.9	80.2	30.8	95.0	62.1	93.2	84.7	79.9	87.1	65.4	97.8	72.4	93.2
Verbal Reinforcement Learning														
Memento (Arxiv'25)	76.6	73.5	75.0	23.1	94.4	62.1	94.1	82.7	79.4	83.3	48.1	95.6	74.1	94.1
Training-Free-GRPO (Arxiv'25)	78.6	75.8	80.7	28.9	95.6	63.8	89.0	85.5	82.6	85.9	67.3	97.8	72.4	91.5
Inference-Time Compute Scaling														
Compute-Optimal (ICLR'25)	80.8	78.1	85.9	28.9	99.0	63.8	86.4	86.9	82.6	89.6	69.2	**100**	77.6	90.7
RGCWM (Ours)	**85.3**	**81.9**	**88.5**	32.7	**100**	**77.6**	**97.5**	**91.2**	**88.2**	**93.2**	**73.1**	**100**	**81.0**	**98.3**

Table 1: Main results on GUIDEBENCH across two base models (Qwen3-8B and DeepSeek-R1). RGCWM consistently outperforms prior approaches across all categories, achieving the best overall performance while maintaining strong robustness on hallucination-sensitive subsets. Best results are shown in bold, and second-best results are underlined.

lected action is unchanged for two consecutive iterations. The rule causal graph $\mathcal{G}$ is constructed once per guideline and kept fixed; propagation updates only z_t with $\beta = 0.5$. When enabled, PCR runs one Critic-Refiner round per iteration. All calls use greedy decoding. Additional settings and analyses are provided in Appendices A.8, A.7, and A.6.

Main Results. Table 1 reports results on GUIDEBENCH with two backbones, Qwen3-8B and DeepSeek-R1. RGCWM achieves the best overall accuracy in both cases (85.3% and 91.2%), outperforming the strongest non-RGCWM baseline (Compute-Optimal) by 4.5% and 4.3%, respectively, under identical inference-time settings. Beyond aggregate accuracy, RGCWM shows pronounced gains on categories most sensitive to rule interactions. On *Price matching*, where prerequisite and exclusion rules are prevalent, RGCWM improves over Compute-Optimal by +3.9% (Qwen3-8B: 81.9% vs. 78.1%) and +5.7% (DeepSeek-R1: 88.2% vs. 82.6%). On *Summarization*, which often requires suppressing partially applicable or conflicting guidelines, the margin further widens (+13.8% on Qwen3-8B and +3.4% on DeepSeek-R1). Most notably, on *Hallucination Detection*, RGCWM reaches near-ceiling performance (97.5% / 98.3%), consistently surpassing all baselines. Across these categories, failures of text-based methods are dominated not by missing local reasoning steps, but by incorrect activation or co-activation of interdependent rules.

The improvements are stable across backbones. Although DeepSeek-R1 is uniformly stronger than Qwen3-8B, RGCWM consistently adds a further 4-5% margin over the strongest backbone-specific baselines. This consistency indicates a backbone-agnostic inference advantage, arising from explicit planning over a dynamic rule-evidence state under a fixed causal structure, rather than backbone-specific prompt sensitivity. In contrast, methods that primarily scale textual reasoning depth (e.g., CoT, Reflexion, ReAct) or deliberative diversity (Debate, Collaboration) exhibit smaller and less uniform gains, particularly on tasks dominated by rule conflicts and prerequisites.

Finally, RGCWM also outperforms inference-time compute scaling (Compute-Optimal), indicating that additional test-time computation alone is insufficient when it is not structured around rule dynamics. RGCWM instead reallocates inference effort to qualitatively different operations: 1-Call Rollout enables counterfactual comparison of candidate actions from a shared world state, while graph-aware scoring explicitly penalizes conflict co-activation and favors globally stable configurations. Consequently, computation is translated into

improved guideline consistency rather than longer but brittle reasoning traces.

Overall, these results demonstrate that explicit causal modeling of guideline structure ($\mathcal{G}$), together with a dynamic rule-evidence state (z_t) and counterfactual planning, yields robust, scalable, and backbone-agnostic improvements under the full-rule inference setting.

7 Ablation Analysis

Component-wise Ablation. Table 2 presents a component-wise ablation under the full-rule setting. Adding the **Static Rule Graph** yields the largest single-module gain, improving accuracy from 77.5% to 80.1% (+2.7%), highlighting the importance of explicitly modeling prerequisite, conflict, and exclusion relations. Building on this structure, **1-Call Rollout** provides a further improvement to 81.4% (+1.3%) via counterfactual screening of candidate actions, while **Graph Propagation** delivers a larger gain (83.1%, +1.7%) by enforcing global consistency across interdependent rules. Finally, enabling **PCR** achieves the best performance (85.3%, +2.2%), indicating that localized refinement complements global planning by correcting residual inconsistencies.

Overall, each component contributes non-redundantly: the rule graph establishes structural validity, rollout supports informed action selection, propagation enforces global coherence, and PCR supplies targeted local repair, together realizing the full benefit of world-model-driven guideline reasoning.

Config	Graph	Rollout	GP	Acc (%)
Baseline (Text-only)				77.5
+ Graph	✓			80.1
+ Graph + Rollout	✓	✓		81.4
+ Graph + Rollout + GP	✓	✓	✓	83.1
Full RGCWM (+ PCR)	✓	✓	✓	**85.3**

Table 2: Ablation study of RGCWM components on GUIDEBENCH (Qwen3-8B, full-rule setting).

Efficiency Analysis. Table 3 compares inference-time latency per instance on Qwen3 under identical hardware and decoding settings. Iterative and multi-agent methods are slower due to repeated LLM calls and growing interaction histories. Debate is particularly expensive (9 calls, 95.6s) yet underperforms RGCWM, whereas RGCWM achieves the best accuracy (85.3%) with moderate cost (4 calls, 32.3s). This efficiency stems from consolidating planning and candidate evaluation into a small number of structured calls, suggesting that RGCWM gains come from structured causal evaluation rather than brute-force test-time scaling.

Method	Calls	Time (s)	Acc (%)
CoT	1.0	6.4	73.5
ReAct	5.0	33.1	76.3
Reflexion	3.0	18.8	75.9
Debate	9.0	95.6	79.6
Collaboration	4.0	25.4	77.6
Full RGCWM	4.0	32.3	85.3

Table 3: Latency (s) versus accuracy (%) on Qwen3 (average wall-clock time per instance; same hardware/decoding).

Weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	γ	**Acc (%)**
Equal (0.25, 0.25, 0.25, 0.25)	1	84.9
Ours (0.40, 0.30, 0.20, 0.10)	1	**85.3**
More-conflict (0.30, 0.40, 0.20, 0.10)	1	85.1
More-stability (0.40, 0.20, 0.30, 0.10)	1	85.0
Less-conf (0.45, 0.25, 0.20, 0.10)	1	84.8
Ours, γ sweep	0.5	85.1
Ours, γ sweep	2.0	85.0

Table 4: Sensitivity of the structural validity score (Eq. 3) to fixed weights and γ.

Weight Sensitivity. Table 4 analyzes the sensitivity of the structural validity score (Eq. 3) to different fixed weight allocations and the conflict sharpness parameter γ. Across a range of reasonable weight configurations, performance remains within a narrow band (84.8%-85.3% Acc.), indicating that RGCWM is robust to moderate reweighting and does not depend on fine-grained score tuning. Similarly, varying γ over a two-fold range (0.5-2.0) results in negligible performance changes. Based on this stability, we adopt a single global configuration λ_1=0.4, λ_2=0.3, λ_3=0.2, λ_4=0.1 and γ=1 for all experiments, without per-task or per-category tuning. This choice emphasizes rule-level consistency while retaining balanced sensitivity to conflict avoidance and state stability. Empirically, the confidence term L_k primarily influences candidate selection when structural scores are close; accordingly, it is assigned a smaller weight. Overall, these results suggest that RGCWM's gains arise from its causal structure and state modeling, rather than from sensitive hyperparameter choices.

Efficiency, Cost, and Robustness Analysis. We further analyze RGCWM from three complementary perspectives: overall inference efficiency, the effect of graph induction source, and module-wise token consumption. Table 5 compares the overall token usage of RGCWM against representa-

Method	Acc (%)	Total Tokens
GuideBench	77.5	≈1.7M
Compute-Optimal	80.8	≈113M
RGCWM	**85.3**	≈7.0M

Table 5: Overall efficiency comparison across methods. RGCWM achieves higher accuracy than compute-optimal reasoning with substantially lower total token usage.

Model	Graph Inducer	Acc (%)	Avg Tokens	Approx. Cost
Qwen3-8B	Text-only (No Graph)	77.5	–	–
Qwen3-8B	Manual (Human)	82.9	–	–
Qwen3-8B	DeepSeek-R1 (API)	84.9	≈ 88K	≈ $0.03
Qwen3-8B	Qwen3-8B (Local)	84.7	≈ 50K	–
Qwen3-8B	GPT-5.2 (API)	85.3	≈ 28K	≈ $0.10

Table 6: Effect of graph induction source on RGCWM performance. The inference backbone is fixed to Qwen3-8B. Performance remains stable across different graph induction sources, indicating that gains are not driven by the inducing model itself.

tive inference-time baselines. RGCWM achieves higher accuracy than compute-optimal reasoning while using substantially fewer tokens (85.3% with ≈7.0M tokens versus 80.8% with ≈113M tokens), indicating that structured state-based planning is more efficient than unstructured test-time scaling. Compared to the text-only baseline, RGCWM requires additional inference-time computation, but translates this computation into more effective rule-consistent reasoning rather than longer free-form reasoning traces.

Table 6 examines the impact of the graph induction source while fixing the inference backbone to Qwen3-8B, thereby isolating the effect of graph construction. Performance remains stable across different model-induced graphs (84.7–85.3), and all graph-based variants substantially outperform the text-only baseline (77.5). This suggests that the gain is not primarily driven by the strength of the graph-inducing model, but by the structured state tracking and graph propagation mechanism enabled by RGCWM. Even a manually constructed graph yields a clear improvement (82.9), further supporting the importance of explicit structure. Graph construction is performed only once per guideline and amortized across all downstream instances.

Table 7 reports the internal token distribution of RGCWM across major modules during the full evaluation. The dominant computation is concentrated in rollout and PCR-based inference, while graph construction accounts for only 0.4% of the total token usage. This shows that the structural components of RGCWM introduce only negligible overhead, and that most computation is spent on

Module	Tokens	Percentage
Graph Construction	≈28,000	0.4%
Rollout (1-call)	≈3,350,000	47.9%
PCR – Proposer	≈1,250,000	17.9%
PCR – Critic	≈850,000	12.1%
PCR – Refiner	≈1,522,000	21.7%
Total	≈7,000,000	100%

Table 7: Module-wise token distribution of RGCWM during the full evaluation. Graph construction accounts for only a negligible fraction of total token usage and is amortized across all instances.

inference-time reasoning rather than offline graph preparation. Together with the robustness analysis in Appendix A, these results support that the performance gain of RGCWM arises from its structural mechanism rather than from costly graph construction.

8 Conclusion

We proposed **RGCWM**, a **R**ule-**G**rounded **C**ausal **W**orld **M**odel that reformulates guideline-following as stateful planning over an explicit causal rule system. By externalizing guidelines into a static Rule Causal Graph and maintaining a dynamic rule-evidence state, RGCWM enables counterfactual evaluation of candidate responses at inference time without additional training. Experiments on GuideBench show consistent improvements over strong baselines across diverse task categories and two backbone models, with the largest gains on tasks governed by prerequisite, conflict, and exclusion relations, where text-based methods tend to fail. RGCWM also achieves these gains with substantially fewer tokens than compute-scaling approaches, confirming that structured causal planning is a more effective use of inference-time computation than unstructured scaling. We hope this work encourages further exploration of explicit rule-state representations as a foundation for robust and controllable guideline reasoning.

Limitations

RGCWM relies on the base LLM to induce the rule causal graph; if the model is weak or miscalibrated, graph noise can reduce the gains from propagation and structural scoring. In practice, however, this impact is typically limited: the core improvements come from explicit rule-evidence state tracking and rollout-based evaluation under the full guideline

set. Thus, even with imperfect relations, RGCWM remains beneficial, and in the worst case degrades gracefully to a non-graph variant; the graph inducer can also be upgraded or replaced by stronger extractors/checkers.

Compared to single-pass prompting, RGCWM introduces additional inference-time cost due to batched rollout scoring and optional refinement. In our implementation, this overhead is controlled by small fixed budgets and remains competitive with (or lower than) many multi-agent deliberation and compute-scaling baselines. Nonetheless, further efficiency improvements, such as caching reusable evidence, reducing repeated state evaluations, and lightweight scoring approximations, remain valuable future work.

Acknowledgments

This work was partially supported by US National Science Foundation IIS-2412195, CCF-2400785, the Cancer Prevention and Research Institute of Texas (CPRIT) award (RP230363), the National Institutes of Health (NIH) R01 award (1R01AI190103-01) and Microsoft Accelerate Foundation Models Research (2024).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, Qinrui Zhu, and Huanhuan Chen. 2025. Llm-driven causal discovery via harmonized prior. *IEEE Transactions on Knowledge and Data Engineering*.

Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, et al. 2025. Training-free group relative policy optimization. *arXiv preprint arXiv:2510.08191*.

Jiaxiang Chen, Zhuo Wang, Mingxi Zou, Qifan Wang, and Zenglin Xu. 2025. Guideline forest: Experience-induced multi-guideline reasoning with stepwise aggregation. *arXiv preprint arXiv:2506.07820*.

Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, et al. 2024. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428*.

Lingxiao Diao, Xinyue Xu, Wanxuan Sun, Cheng Yang, and Zhuosheng Zhang. 2025. Guidebench: Benchmarking domain-oriented guideline following for llm agents. *arXiv preprint arXiv:2505.11368*.

Guanting Dong, Xiaoshuai Song, Yutao Zhu, Runqi Qiao, Zhicheng Dou, and Ji-Rong Wen. 2024. Toward general instruction-following alignment for retrieval-augmented generation. *arXiv preprint arXiv:2410.09584*.

Noah D Goodman, Vikash K Mansinghka, and Joshua B Tenenbaum. 2007. Learning grounded causal models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29.

Luke Guerdan, Amanda Coston, Zhiwei Steven Wu, and Kenneth Holstein. 2023. Ground (less) truth: A causal framework for proxy labels in human-algorithm decision-making. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 688–704.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2020. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. 2025. Multi-agent debate for llm judges with adaptive stability detection. *arXiv preprint arXiv:2510.12697*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*.

Jiayi Kuang, Ying Shen, Jingyou Xie, Haohao Luo, Zhe Xu, Ronghao Li, Yinghui Li, Xianfeng Cheng, Xika Lin, and Yu Han. 2025. Natural language understanding and inference with mllm in visual question answering: A survey. *ACM Computing Surveys*, 57(8):1–36.

Tobias Lang and Marc Toussaint. 2010. Planning with noisy probabilistic relational rules. *Journal of Artificial Intelligence Research*, 39:1–49.

Mufei Li, Siqi Miao, and Pan Li. 2024. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. 2024. Chain-of-action: Faithful and multimodal question answering through large language models. *arXiv preprint arXiv:2403.17359*.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. *arXiv preprint arXiv:2310.05066*.

Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason Weston, and Jane Yu. 2024. Self-consistency preference optimization. *arXiv preprint arXiv:2411.04109*.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv preprint arXiv:2310.03668*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Wangtao Sun, Chenxiang Zhang, XueYou Zhang, Xuanqing Yu, Ziyang Huang, Haotian Xu, Shizhu He, Jun Zhao, and Kang Liu. 2025. Beyond instruction following: Evaluating inferential rule following of large language models. In *China National Conference on Chinese Computational Linguistics*, pages 408–434. Springer.

Yuhao Wang, Chandler Squires, Anastasiya Belyaeva, and Caroline Uhler. 2018. Direct estimation of differences in causal graphs. *Advances in neural information processing systems*, 31.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xiangyu Wen, Jianyuan Zhong, Zhijian Xu, and Qiang Xu. 2025. Guideline compliance in task-oriented dialogue: The chained prior approach. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6750–6776.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Cong Zhang, Xin Deik Goh, Dexun Li, Hao Zhang, and Yong Liu. 2025. Planning with multi-constraints via collaborative language agents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10054–10082.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, et al. 2025. Memento: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv:2508.16153*.

A Appendix: Reliability and Reproducibility Details

A.1 Robustness and Reproducibility under Estimation Noise

The rule causal graph $\mathcal{G}$ is inferred by a frozen LLM and may contain noisy or imperfect dependency relations. To evaluate robustness under such estimation noise, we conduct controlled perturbation experiments on $\mathcal{G}$ while keeping all other components fixed.

Specifically, we simulate graph noise by randomly removing a proportion p of edges from $\mathcal{G}$ (edge dropout), with $p \in \{20\%, 30\%\}$. This perturbation disrupts both supportive and conflicting dependencies and serves as a stress test for graph reliability.

Graph Condition	Acc. (%)
Full graph	85.3
– 20% edges	84.1
– 30% edges	82.9
Text-only baseline	77.5

Table 8: Performance under rule causal graph perturbations (Qwen3-8B).

Performance degrades smoothly as graph quality decreases and remains consistently above the text-only baseline, indicating graceful degradation rather than systematic failure under imperfect graph construction.

A.2 Robustness to Rule-Evidence Estimation Noise

In addition to graph structure, RGCWM relies on model-induced rule-evidence estimates $\hat{z}_t^{(k)}$ produced by the frozen LLM during the 1-Call Rollout. To test robustness to noise in these estimates, we inject zero-mean Gaussian noise into $\hat{z}_t^{(k)}$ at inference time: $\tilde{z}_t^{(k)} = \text{clip}(\hat{z}_t^{(k)} + \epsilon)$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where clipping is applied to keep values in $[0, 1]$. All other components (graph, propagation, PCR, and termination) are kept identical.

We observe a gradual degradation as σ increases, while performance remains consistently above the Graph+Rollout (w/o propagation) baseline, indicating that RGCWM does not rely on finely calibrated rule-evidence values.

A.3 Rule-Evidence Extraction in 1-Call Rollout

RGCWM represents the dynamic guideline state as a continuous rule-evidence vector $z_t \in [0, 1]^{|\mathcal{R}|}$,

Evidence Noise Level	Acc. (%)
$\sigma = 0$ (no noise; full RGCWM)	85.3
$\sigma = 0.05$	84.6
$\sigma = 0.10$	83.8
$\sigma = 0.20$	82.2
Graph+Rollout (w/o GP)	81.4
Text-only baseline	77.5

Table 9: Robustness to noise in rollout-induced rule-evidence estimates $\hat{z}_t^{(k)}$ on GUIDEBENCH (Qwen3-8B, full-rule).

where each entry indicates evidence that a rule is applicable and satisfied under the current trajectory. Importantly, low evidence does not uniquely imply violation: it can also arise from inapplicability or insufficient information in the context.

Joint structured evaluation. At rollout time, we evaluate each candidate action $a_t^{(k)}$ jointly against the full rule set $\mathcal{R}$ under the same world state (h_t, z_t). Concretely, the 1-Call Rollout prompts the frozen LLM to output a structured object containing: (i) a per-rule evidence vector $\hat{z}_t^{(k)}$ with values in $[0, 1]$, and (ii) a lightweight confidence cue L_k (defined in App. A.4). This evaluation is performed in the same rollout call used for candidate screening, without additional LLM queries.

Rubric for evidence scores. For each rule r_i, the model is instructed to assign $\hat{z}_t^{(k)}(i)$ based on:

- **1.0:** clearly applicable and satisfied by $a_t^{(k)}$ under the provided context;
- **0.5:** unclear / partially supported / insufficient context to decide applicability;
- **0.0:** clearly violated, or clearly inapplicable given the context and prerequisites.

Intermediate values are allowed to express graded evidence when the satisfaction signal is partial. We initialize $z_0 = 0.5$ as a uniform prior, corresponding to an uninformative starting state.

Parsing and normalization. We parse the structured rollout output to extract $\hat{z}_t^{(k)}$ and L_k, clip values to $[0, 1]$, and align the evidence vector to the fixed rule indexing of $\mathcal{R}$. All subsequent scoring and propagation operate only on these normalized vectors.

A.4 Confidence Cue L_k: Definition and Usage

In addition to rule-evidence vectors, the rollout returns a scalar confidence cue $L_k \in [0, 1]$ for each candidate. L_k is self-reported by the same frozen LLM as part of the structured rollout output, intended only as a lightweight preference signal

reflecting the model's internal certainty about the candidate's overall compliance.

Not calibrated uncertainty. We explicitly do not treat L_k as a calibrated probability of correctness, and we do not use token log-probabilities. Accordingly, L_k is assigned a small weight in Eq. (7) and functions primarily as a tie-breaker when structural scores are close.

No extra compute. L_k is produced in the same 1-Call Rollout that produces $\hat{z}_t^{(k)}$, requiring no additional LLM calls beyond those already counted in the rollout.

A.5 Analysis of the Structural Validity Score

We further analyze the structural validity score in Eq. 3 by ablating individual terms. Unless otherwise stated, we evaluate the full RGCWM pipeline end-to-end (including propagation, PCR, and termination), and only modify the score used in RolloutSelect. When ablating a term, we set its coefficient to zero and keep all remaining coefficients unchanged (no renormalization) to isolate its contribution under the same scale.

Score Variant	Acc. (%)
Full score (Eq. 3)	85.3
w/o conflict term (λ_2=0)	83.5
w/o stability term (λ_3=0)	83.2
w/o both (λ_2=λ_3=0)	81.1
w/o L_k (λ_4=0)	85.0

Table 10: Ablation of individual components in Eq. 3 on GuideBench (Qwen3-8B, full-rule).

Removing either the conflict penalty or the stability term results in a clear drop, and removing both leads to a larger degradation, indicating that the two terms capture related but non-redundant signals. In contrast, removing L_k yields a small decrease, consistent with L_k acting as a weak tie-breaker rather than a primary driver.

A.6 Decoding, Backbones, and Determinism

All experiments are conducted at inference time using frozen LLM backbones. We use **Qwen3-8B** and **DeepSeek-R1** as the primary reasoning models for all task-level inference, without any fine-tuning or parameter updates.

Graph inference model and scope. We use **GPT-5.2** exclusively to infer the static rule causal graph $\mathcal{G}$ from the guideline rule set S, i.e., to predict pairwise rule relations τ_{ij} and confidence scores c_{ij}. Graph construction is performed once per guideline specification and cached; $\mathcal{G}$ is then fixed for all downstream reasoning steps and all test instances under the same guideline. Importantly, GPT-5.2 is not used for task-level reasoning, candidate generation, rollout evaluation, Critic, or Refiner. All per-instance inference uses the evaluated backbone (Qwen3-8B or DeepSeek-R1). In call/token accounting, graph construction cost is reported separately from per-instance inference cost.

Decoding configuration. To ensure determinism and fair comparison across methods, all LLM calls (including graph inference, 1-Call Rollout, Critic, and Refiner) use greedy decoding with temperature $= 0$, top-p $= 1.0$, and top-k disabled. The maximum generation length is fixed across methods and tasks. No stochastic sampling is used during decoding.

Determinism for graph construction. Although graph construction aggregates over M=3 independent LLM calls (as described in Sec. 5) to reduce variance, each call uses greedy decoding. Thus, the aggregation reflects repeated model queries rather than sampling-based decoding. Any tie-breaking in post-processing (e.g., majority voting) is deterministic. Robustness to graph perturbations is further analyzed in Appendix A.1.

Randomness control. Since all decoding is greedy, the system is deterministic given the same guideline specification and inputs. The only sources of randomness appear in explicit robustness experiments (e.g., graph edge dropout or rule-evidence noise), where random seeds are fixed and reported.

Batching and efficiency. Candidate actions in the 1-Call Rollout are evaluated jointly in a single batched LLM call, ensuring identical context and decoding settings for all candidates and avoiding any bias from sequential evaluation.

A.7 History Abstraction and Update Mechanism

RGCWM maintains a compact history abstraction h_t to summarize prior reasoning steps and corrective feedback, enabling stateful planning without unbounded context growth.

Representation. The history state h_t is a short natural-language summary capturing: (i) previously selected actions, (ii) identified rule violations or conflicts, and (iii) key corrective constraints introduced by the Critic–Refiner. It does not store raw intermediate generations or full trajectories. The length of h_t is capped to a fixed budget to ensure stable context size.

Model	Params	Access	Identifier / Version	Provider
GPT-5.2	undisclosed	API	gpt-5.2-2025-10-01	OpenAI
DeepSeek-R1	671B	API	deepseek-r1-2025-05-28	DeepSeek
Qwen3-8B	8B	weights	official release	Qwen

Table 11: Backbone models and API identifiers used in our experiments.

β	Acc. (%)
0.3	84.9
0.4	85.1
0.5	**85.3**
0.6	85.2
0.7	85.0
0.8	84.8

Table 12: Sensitivity to step size β.

Update rule. At each iteration, if the Critic–Refiner module is enabled, the Refiner produces a concise correction summary describing how the selected action should be adjusted to better satisfy the guideline. This summary is appended to the existing history and then re-summarized into a compact form to produce h_{t+1}. If PCR is disabled or no refinement is triggered, the history remains unchanged ($h_{t+1} = h_t$).

Determinism and compute budget. History summarization uses the same LLM and greedy decoding configuration as other components. No additional LLM calls beyond those already counted for the Refiner are introduced. Thus, h_t does not alter the call or token budget relative to baselines.

A.8 Propagation Step Size

The propagation step size β in Eq. (8) controls the strength of graph-based state updates. In all experiments, we fix $\beta = 0.5$ globally across tasks and models, without any tuning. To verify robustness, we sweep β in the range $[0.3, 0.8]$ on GUIDEBENCH (Qwen3-8B, full-rule setting). Results are reported in Table 12. Performance varies smoothly across this interval, with no instability observed.

B Appendix: Full Pseudocode of RGCWM Inference

The full Pseudocode is provided in Algorithm 2.

C Appendix: Prompt Templates

Figures 3–8 present abridged excerpts of the prompt templates used in RGCWM, covering the world state representation, rule graph specification, and the four-stage inference pipeline. These figures are intended to illustrate the shared structural design and role separation of the prompts, rather than to provide the full templates used during inference. For clarity and space constraints, each figure shows a shortened, representative snapshot that highlights the core fields and control logic common across task domains.

Algorithm 2 RGCWM Inference (Full Version)

Require: Query x_q, guideline rules $S = \{r_i\}_{i=1}^{|\mathcal{R}|}$, fixed graph $\mathcal{G} = (\mathcal{R}, \mathcal{E}, \mathcal{W})$
Require: Frozen backbone LLM L; hyperparameters $K, T_{\max}, \beta, \tau_c, \tau_\Delta, K_{\text{stable}}$
Note. $\mathcal{G}$ is constructed offline once per guideline and cached. All components (PROPOSER, ROLLOUTEVAL, EVIDENCEEVAL, CRITICREFINER, SUMMARIZE) share the same frozen backbone L. $\text{clip}(u) = \min(1, \max(0, u))$ is applied element-wise, and $\text{C}(z) = \frac{1}{|\mathcal{R}|}\sum_i z(i)$.
$h_0 \leftarrow \emptyset$
$z_0 \leftarrow 0.5 \cdot \mathbf{1} \in [0,1]^{|\mathcal{R}|}$
$\tilde{a}_{-1} \leftarrow \emptyset$
$\text{Score}_{-1}^{\text{struct}} \leftarrow -\infty$
stable $\leftarrow 0$
for $t = 0, 1, \ldots, T_{\max} - 1$ **do**
  (A) Propose K candidate actions
  $\mathcal{A}_t \leftarrow \text{PROPOSER}(x_q, h_t, z_t, S; L)$
  (B) 1-Call Rollout: joint evaluation
  $\{(\hat{z}_{t+1}^{(k)}, L_k)\}_{k=1}^{K} \leftarrow \text{ROLLOUTEVAL}(x_q, h_t, z_t, \mathcal{A}_t, S; \mathcal{G}, L)$
  $\text{Score}_k \leftarrow \text{STRUCTSCORE}(\hat{z}_{t+1}^{(k)}, L_k, \mathcal{G})$ for all k
  $k^\star \leftarrow \arg\max_k \text{Score}_k$
  $\tilde{a}_t \leftarrow \mathcal{A}_t[k^\star]$
  $\hat{z}_{t+1}^\star \leftarrow \hat{z}_{t+1}^{(k^\star)}$
  $\text{Score}_t^{\text{struct}} \leftarrow \text{Score}_{k^\star}$
  $\text{fb}_t \leftarrow \emptyset$
  (C) Optional PCR refinement (no re-planning)
  if USEPCR **then**
    $(\tilde{a}_t, \text{fb}_t) \leftarrow \text{CRITICREFINER}(x_q, h_t, \tilde{a}_t, S; L)$
    $\hat{z}_{t+1}^\star \leftarrow \text{EVIDENCEEVAL}(x_q, h_t, z_t, \tilde{a}_t, S; L)$
    $\text{Score}_t^{\text{struct}} \leftarrow \text{STRUCTSCOREONLY}(\hat{z}_{t+1}^\star, \mathcal{G})$
  end if
  (D) Graph propagation
  $\Delta z_t \leftarrow \hat{z}_{t+1}^\star - z_t$
  for $j = 1, 2, \ldots, |\mathcal{R}|$ **do**
    $z_{t+1}(j) \leftarrow \text{clip}\Big(\hat{z}_{t+1}^\star(j) + \beta \sum_{i \in \text{InNbr}(j)} w_{ij}\, \Delta z_t(i)\Big)$
  end for
  (E) Update history
  $h_{t+1} \leftarrow \text{SUMMARIZE}(h_t, \tilde{a}_t, \text{fb}_t)$
  (F) Termination checks
  if $\text{C}(z_{t+1}) \geq \tau_c$ **then**
    return $\tilde{a}_t$
  end if
  if $t \geq 1$ **and** $\left|\text{Score}_t^{\text{struct}} - \text{Score}_{t-1}^{\text{struct}}\right| \leq \tau_\Delta$ **then**
    return $\tilde{a}_t$
  end if
  if $\tilde{a}_t = \tilde{a}_{t-1}$ **then**
    stable $\leftarrow$ stable $+ 1$
  else
    stable $\leftarrow 0$
  end if
  if stable $\geq K_{\text{stable}}$ **then**
    return $\tilde{a}_t$
  end if
  $z_t \leftarrow z_{t+1}$; $h_t \leftarrow h_{t+1}$
  $\tilde{a}_{t-1} \leftarrow \tilde{a}_t$
  $\text{Score}_{t-1}^{\text{struct}} \leftarrow \text{Score}_t^{\text{struct}}$
end for
return $\tilde{a}_{T_{\max}-1}$

You are a rule causal graph constructor.
[ROLE] Your task is to infer directed causal relations between guideline rules. You do NOT solve the task described by the rules. You do NOT generate answers, actions, responses, or dialogue.

[SCOPE] You ONLY analyze how enforcing one rule affects the applicability or validity of another rule. Each rule must be treated as an atomic unit. All reasoning must be performed strictly at the rule level, not at the level of specific user queries, contexts, examples, or downstream system behavior.

[DIRECTIONALITY] Relations are directional. The relation from rule_i to rule_j may differ from the relation from rule_j to rule_i. You must evaluate each ordered rule pair independently.

[RELATION SET] For each ordered rule pair, you must choose exactly ONE relation label from the following closed set: support, prerequisite, conflict, exclusion, or none. The semantics are strictly defined as follows: support means enforcing rule_i increases the likelihood that rule_j is applicable or satisfied; prerequisite means rule_i must be satisfied before rule_j can apply; conflict means enforcing rule_i makes rule_j harder to satisfy but both rules may still partially apply; exclusion means enforcing rule_i strictly invalidates rule_j; none means no meaningful causal relation exists. If the relation is weak, ambiguous, or uncertain, you must choose none.

[CONSTRAINTS] Do not assume any external knowledge beyond the provided rule texts. Do not reason about downstream answers, system actions, or user intent beyond what is explicitly encoded in the rules. Do not infer transitive or indirect relations; evaluate only the direct causal influence from rule_i to rule_j.

[CONFIDENCE] You must assign a confidence score in the range [0,1], reflecting how strongly the inferred relation is implied by the rule texts alone.

[OUTPUT FORMAT] Output must be a single valid JSON object and nothing else. Do not include explanations, comments, or natural language outside the JSON. The JSON schema must be exactly: {"rule_i": <int>, "rule_j": <int>, "relation": "support | prerequisite | conflict | exclusion | none", "confidence": <float>}.

[INPUT] Below is the full guideline rule set, where each rule has a unique rule_id and a textual description. You will be asked to infer the relation for a specific ordered pair (rule_i, rule_j) using only this information.

Guideline rules: [INSERT FULL RULE LIST HERE].

[TASK] Infer the causal relation from rule_i = i to rule_j = j and output the JSON result following the schema exactly.

Figure 3: **Rule graph prompt (abridged excerpt).** An abridged sample of the prompt that specifies the static guideline causal graph, encoding dependencies, conflicts, and exclusions among rules. The displayed content is a shortened excerpt intended to illustrate the prompt structure rather than the complete graph specification.

<WorldState>

<HistorySummary>
A concise summary of previously confirmed decisions, refinements, and resolved rule constraints. This summary constrains future reasoning and prevents contradiction
with already accepted conclusions.
</HistorySummary>

<RuleEvidence>
The current rule-evidence state z_t. It is a vector of length N, where N equals the total number of rules. The order strictly follows the order of the rule list.
</RuleEvidence>

<RuleEvidenceVector>
[z_1, z_2, ..., z_N]
</RuleEvidenceVector>

<RuleEvidenceSemantics>
- 1.0 : The rule is clearly applicable and satisfied
- 0.5 : Applicability or satisfaction is uncertain
- 0.0 : The rule is clearly violated or not applicable
</RuleEvidenceSemantics>

<GraphEvidence>
A localized summary of rule dependencies extracted from a fixed rule graph, including support, prerequisite, conflict, or exclusion relations
relevant to the current reasoning context. This information is read-only and does not require graph reconstruction.
</GraphEvidence>

</WorldState>

Figure 4: **World State prompt (abridged excerpt).** A shortened illustrative snapshot of the prompt used to represent the current world state in RGCWM, including the rule evidence vector, history summary, and graph-level evidence. The figure shows a condensed excerpt for visualization purposes only and does not contain the full prompt used during inference.

```
Your role is to generate multiple candidate answers under a fixed rule-constrained environment.

Your objective:
Given the task instruction, full guideline set, context, query, and the current WorldState, produce several complete and executable candidate answers to be evaluated in later stages.

Constraints:
1. This stage only generates candidates; no evaluation is allowed.
2. Do not analyze or reference rule satisfaction.
3. Do not modify or summarize history.
4. Each candidate must be a valid final answer on its own.

Inputs:
<Instruction>...</Instruction>
<Guidelines>...</Guidelines>
<Context>...</Context>
<Query>...</Query>
<WorldState>...</WorldState>

Output format (no extra text):

<ProposedCandidates>
Candidate 1: ...
Candidate 2: ...
Candidate 3: ...
</ProposedCandidates>
```

Figure 5: **Proposer prompt (abridged excerpt).** A condensed excerpt of the prompt used by the Proposer to generate a bounded set of candidate answers conditioned on the current world state and guideline rules. Only a shortened illustrative sample is shown, not the full prompt used in the system.

```
Your role is to evaluate how each candidate answer would affect the rule system under the current world state.

Your objective: For each candidate answer, predict the resulting evidence level for every rule if that candidate were adopted.

Key interpretation: You are evaluating state consequences, not answer quality.

Inputs:
<Instruction>...</Instruction>; <Guidelines>...</Guidelines>;  <Context>...</Context>;  <Query>...</Query>;  <WorldState>...</WorldState>

<Candidates>
Candidate 1: ... ; Candidate 2: ...; Candidate 3: ...
</Candidates>

Evaluation rules: - Output one evidence value per rule (order must match the rule list) - Evidence values must be in [0,1] with fixed semantics:
  - 1.0: clearly applicable and satisfied
  - 0.5: uncertain applicability or satisfaction
  - 0.0: clearly violated or inapplicable

You must consider: - Current RuleEvidenceVector – HistorySummary - GraphEvidence (dependencies, conflicts, exclusions). Additionally, provide a
confidence score L ∈ [0,1] representing your overall confidence in the rule-consistency assessment (not a probability).

Output strictly in JSON, no explanations:
{
  "CandidateEvaluations": [
    {
      "CandidateID": 1,
      "RuleEvidence": [z1, z2, ..., zN],
      "Confidence": L1
    },
    ...
  ]
}
```

Figure 6: **Rollout prompt (abridged excerpt).** A condensed excerpt of the prompt used during inference-time rollout to evaluate multiple candidate actions in parallel and update the world state accordingly. The figure shows a shortened snapshot for visualization and does not include the full rollout prompt. Although the downstream tasks span heterogeneous domains (e.g., mathematical problem solving and multiple-choice selection), RGCWM employs a shared Proposer by abstracting actions as language-level candidate responses rather than domain-specific decisions. The Proposer follows a fixed input–output contract: given the query, guideline set, and current world state, it produces a bounded set of textual candidates. Domain-specific differences are specified through the instantiation of these inputs (e.g., task context formatting, query formulation, and guideline content). Consequently, the proposal mechanism itself remains unchanged across domains, while task semantics are conveyed via the provided inputs.

```
Your role is to audit a given answer for remaining rule-related risks under the current world state.

Your objective:
Identify whether the answer still poses potential rule violations, dependency failures, or ambiguity-induced risks.

Inputs:
<Answer>...</Answer>
<Guidelines>...</Guidelines>
<WorldState>...</WorldState>

Focus on:
1. Explicit rule violations
2. Dependency or prerequisite failures implied by the rule structure
3. Ambiguities that may cause rule misuse

If no modification is needed, explicitly state so.

Output only structured feedback:

{
  "IssuesDetected": true / false,
  "Critique": "..."
}
```

Figure 7: **Critic prompt (abridged excerpt).** A shortened excerpt of the prompt used by the Critic to evaluate the rule-level consequences of each candidate answer under the current world state. The figure presents a condensed snapshot for illustration purposes only.

```
Your role is to minimally revise an answer to reduce identified rule risks while preserving its original intent and structure.

Your objective:
Based on the critique, apply the smallest necessary modification to improve rule consistency, then re-estimate the local rule impact.

Inputs:
<OriginalAnswer>...</OriginalAnswer>
<Critique>...</Critique>
<WorldState>...</WorldState>

Revision principles:
1. Modify only if issues were detected.
2. Keep changes local and minimal.
3. Do not introduce new decision paths or scope expansion.

Output all three components (required):

{
  "RefinedAnswer": "...",
  "UpdatedRuleEvidence": [z1, z2, ..., zN],
  "HistoryUpdate": "Concise summary of the revision and its rule impact"
}
```

Figure 8: **Refiner prompt (abridged excerpt).** An abridged excerpt of the prompt used by the Refiner to apply minimal local revisions that reduce identified rule risks while preserving the original answer structure. The content shown is a shortened illustrative example rather than the complete prompt.