RIG: Synergizing Reasoning and Imagination in End-to-End Generalist Policy

Zhonghan Zhao 1,2* Wenwei Zhang 2* Haian Huang 2 Kuikun Liu 2 Jianfei Gao 2 Gaoang Wang 1† Kai Chen 2†

¹Zhejiang University ²Shanghai AI Laboratory

{zhonghan.22, gaoangwang}@intl.zju.edu.cn, {zhangwenwei,chenkai}@pjlab.org.cn

Abstract

Reasoning before action and imagining potential outcomes (i.e., world models) are essential for embodied agents operating in complex open-world environments. Yet, prior work either incorporates only one of these abilities in an end-toend agent or integrates multiple specialized models into an agent system, limiting the learning efficiency and generalization of the policy. Thus, this paper makes the first attempt to synergize Reasoning and Imagination in an end-to-end Generalist policy, termed RIG. To train RIG in an end-toend manner, we construct a data pipeline that progressively integrates and enriches the content of imagination and reasoning in the trajectories collected from existing agents. The joint learning of reasoning and next image generation explicitly models the inherent correlation between reasoning, action, and dynamics of environments, and thus exhibits more than 17× sample efficiency improvements and generalization in comparison with previous works. During inference, RIG first reasons about the next action, produces potential action, and then predicts the action outcomes, which offers the agent a chance to review and self-correct based on the imagination before taking real actions. Experimental results show that the synergy of reasoning and imagination not only improves the robustness, generalization, and interoperability of generalist policy but also enables test-time scaling to enhance overall performance.

1. Introduction

To navigate the complexities of open-world environments, two quintessential human faculties are *de facto* to embodied agents: imagination of prospective outcomes and reasoning. Although reasoning endows agents with the ability to deconstruct task objectives into executable plans through logical inference, it inherently operates within the constraints of perceptual history. This limitation underscores the complementarity of world models that learn the environmental dynamics, which not only allows the agent to predict ac-

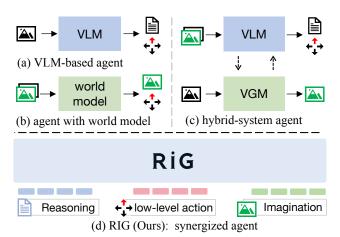


Figure 1. Comparison between conventional agents and RIG. RIG produces reasoning, actions, and imagination within a single Transformer.

tion consequences but also facilitates risk-aware decisionmaking by evaluating hypothetical trajectories.

The synergistic integration of reasoning and imagination constitutes an indispensable foundation for more intelligent and robust embodied agents operating in dynamically evolving environments. However, these two abilities are typically implemented in separate models. Specifically, reasoning mainly exists in large vision language models (VLMs) [33, 35, 43, 44] that parse visual input and produce textual insights and actions (Fig. 1(a)), which lack explicit future prediction mechanisms. In contrast, world models [16, 21] specialize in predicting future frames from video data (Fig. 1(b)), which suffer from data inefficiency due to the implicit learning of concepts, physical laws, and environment dynamics. Recent attempts [42, 45, 48] combine reasoning and imagination by connecting VLMs and visual generative models (VGMs). Yet, the integrated system (Fig. 1(c)) prevents end-to-end optimization of the agent, leaving the mutual benefits between reasoning and world models underexplored.

To bridge these gaps, this paper makes the first attempt to synergize **R**easoning and **I**magination in an end-to-end **G**enralist policy, termed **RIG** (Fig. 1(d)), RIG learns tex-

tual reasoning, low-level action control, and image generation through the sequence-to-sequence modeling objective within an autoregressive Transformer, as we hypothesize that the explicit modeling of the logic and motivation behind actions and their consequences could make RIG capture open-world dynamics more comprehensively and improve the sample efficiency of training.

We develop RIG by adopting a progressive data collection strategy because existing datasets typically lack trajectories that contain interleaved image observations, precise actions, and high-quality textual reasoning. Based on initial trajectories collected from humans [11] and existing agents [20] that contain only actions and image frames, we first use VLM to insert textual rationales before each action on the trajectory and train RIG-basic with the reasoning-enriched trajectories. During inference, RIG-basic generates actions purely from textual and visual inputs, without leveraging imagined future frames, as decisions are executed immediately based on current observations.

To further leverage visual imagination in reasoning to further improve the robustness of the policy, we collect unsuccessful trajectories from RIG-basic and adopt GPT-40 to review and revise these trajectories. Then, the suboptimal trajectories are taken as dreamed trajectories and combined with their corresponding revisions to form dream-review style trajectories for training RIG (also noted as RIGlookahead for clarity). In contrast to RIG-basic that conduct reasoning without imagination, RIG-lookahead learns to first generate a trajectory by taking the predicted images as the environment states, and then review the hypothetical trajectory in reasoning, and predict revised action that changes the environment. Such a design provides scalability at inference time, where the number of steps in the dream trajectory can be scaled so that the agent can more comprehensively understand the effectiveness of the action and make future-aware decisions.

We extensively evaluate RIG in the diverse, open-world Minecraft environment. Experimental results show that RIG upgrades the state-of-the-art results on embodied tasks, image generation, and reasoning benchmarks by $3.29\times$, $2.42\times$, and $1.33\times$, respectively. Such a superior performance is achieved by training RIG on only **111 hours** of videos, which is **17**× fewer than previous works that rely on 2000 hours of videos. Moreover, when scaling the training data, environmental interactions, and the lookahead steps during reasoning, the generalization ability and robustness of RIG consistently improve, which implies the potential of synergizing reasoning and imagination in embodied agents. Our main contributions are summarized as follows:

- We introduce an end-to-end generalist policy that synergistically integrates explicit reasoning and visual imagination.
- We propose a progressive data collection strategy coupled

- with straightforward language model-based training to efficiently implement our method.
- Our method naturally supports test-time scaling, enabling dynamic lookahead reasoning that enhances action robustness and reduces trial-and-error during inference.

2. Related Work

Embodied Agents in Minecraft. Minecraft presents a significantly open-ended and complex environment [6, 11, 13, 18, 36] for embodied agents. Early approaches leveraged explicit world models to predict future states [7, 16] but lack textual reasoning capabilities. Inspired by large language models (LLMs) [5, 32], subsequent methods combined LLMs with low-level controllers to address longhorizon tasks. For example, Voyager [33] and STEVE [43] used LLMs for high-level planning integrated with code databases, while others like Jarvis-1 [35] paired LLMs with pre-trained low-level policy models such as VPT [3]. However, these methods typically lack a world model to explicitly anticipate future visual outcomes. More recently, MineDreamer [48] integrates a world model and a policy controller, yet treats vision generation and policy control as separate modules, limiting coherent multi-modal reasoning. In contrast, RIG first attempts to explore an end-to-end generalist policy that simultaneously learns textual reasoning, visual imagination, and low-level action predictions to achieve high generalization ability and sample efficiency.

World Models for Embodied Agents. Learning robust world models is essential for embodied agents to effectively plan and act within simulated environments [19, 27]. Early approaches primarily focused on action-conditioned video prediction or latent imagination for sample-efficient rollouts [14, 15, 17, 21, 30], yet they often tightly coupled the world model with specific policies, limiting their adaptability. Inspired by recent successes in large-scale pretraining [24, 37] and Transformer-based architectures [25], several methods now leverage generalizable knowledge to model visual and textual distributions. However, these models typically overlook explicit reasoning and deeper causal relationships between actions and resulting visual states. RIG explicitly learns to model the joint distribution of textual reasoning, actions, and their visual consequences to enable more accurate predictions of complex and evolving environment dynamics.

Unified Understanding and Generation. Multi-modal Large Language models (MLLMs) aim to tackle understanding and generation tasks across different modalities [22, 47] within a unified architecture. Existing methods typically train on large-scale image-text datasets to improve general visual understanding and generation capabilities [34, 38, 39, 47]. However, these datasets lack the interleaved action and reasoning trajectories required for training embodied agents, limiting their direct applicability

to real-world embodied scenarios. Generalist policies like GATO [29] and RT-1 [4] demonstrate multitask capabilities but optimize each task individually without fully leveraging inter-modal synergies. Our work synergizes textual reasoning, low-level action predictions, and visual generation.

3. Method

This paper makes the first attempt to explore the synergy of **R**easoning and **I**magination in an end-to-end **G**eneralist policy, termed RIG. RIG models image, textual reasoning, and textual action in a sequence-to-sequence manner (§ 3.1). We adopt a progressive data collection strategy to first obtain RIG-*basic* that can reason before action but without imagination (§ 3.2), then approach RIG-*lookahead* that learns to reason based on generated trajectories (§ 3.3).

3.1. Preliminary

Typical generalist policies follow an autoregressive paradigm to predict actions based on observations [16, 21]. RIG extends this framework by explicitly generating intermediate textual reasoning before action prediction. Specifically, given multi-modal inputs $X = \{x^{\rm IMG}, x^{\rm TXT}\}$ comprising visual tokens $x^{\rm IMG}$ and textual tokens $x^{\rm TXT}$, RIG learns to autoregressively generate textual reasoning tokens Y, low-level action tokens A, and visual prediction tokens P:

$$(Y, A, P) = \mathcal{F}(X), \quad X = \{x^{\text{IMG}}, x^{\text{TXT}}\}.$$
 (1)

The model is trained in an end-to-end manner using only cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1} \log P_{\theta}(x_i \mid x_{< i}). \tag{2}$$

where $P_{\theta}(\cdot \mid \cdot)$ denotes the conditional probability distribution parameterized by the weights θ of RIG.

3.2. Reasoning without Imagination

Our primary goal is to develop a synergized model capable of simultaneously generating textual reasoning, precise low-level actions, and visual outcome predictions. Since existing agents mainly produce actions, existing accessible datasets typically lack comprehensive trajectories containing all these elements. Therefore, we propose a progressive data collection strategy to gradually enrich these elements in accessible agentic trajectories. Inspired by the recent success of vision-language models [28] that can conduct chain-of-thought (CoT) reasoning given images, our first step is to add reasoning into the action-image trajectories using VLMs to obtain RIG-basic that can conduct reasoning before action.

Data Collection (S0–S2). As shown in Fig. 2, we first refine or collect data from relabeled human play trajectories (S0) and specialized policies (S1), unify their formats, and

add reasoning contents before each action (S2). The details are as below:

- S0 (Refined MineRL-V0): We use trajectories from MineRL-V0 [13] and quantize the camera actions of the original trajectory into discrete 5-degree intervals and then represent them as textual tokens. All other discrete low-level actions retain their original semantic labels.
- **S1** (**Vision-Action, 446K**): We use a pretrained policy, STEVE-1 [20], to collect high-resolution (384×384) image-action pairs and ensure precise visual-action alignment for learning low-level control.
- S2 (Vision-Reasoning, 200K): To integrate reasoning in the original trajectories, we employ GPT-40 as a Reasoner to annotate explicit textual rationales conditioned on visual observations x^{IMG} and the corresponding low-level actions A, formed as $Y = \text{Reasoner}(x^{\text{IMG}}, A)$.

All these trajectories are rigorously filtered based on task success, diversity across environment seeds, and manual validation of reasoning quality. We train RIG-basic using datasets obtained from S0, S1, and S2.

Reasoning without Imagination. After training on datasets (S0, S1, S2), the resulting model, RIG-basic naturally supports multi-round interactions with the environment. As shown in Fig. 3, at each step, it autoregressively generates textual CoT reasoning Y, low-level actions A, and action outcomes P:

$$(Y_{i+1}, A_{i+1}, P_{i+1}) \stackrel{\mathcal{F}}{\leftarrow} (X_i, Y_i, A_i). \tag{3}$$

This unified approach achieves significantly better generalization than traditional methods, requiring substantially fewer training samples (Fig. 4).

3.3. Lookahead Reasoning

Although RIG-basic demonstrates strong baseline performance, the reasoning is still purely based on the perceptual history and does not fully exploit the generative imagination capabilities. To address this, we further augment our datasets with reflective reviewing annotations in stages 3 and 4 (S3 and S4 in Fig. 2), to endow the model with the ability to conduct *lookahead* reasoning, *i.e.*, internally simulate imagined trajectories first, and then take actions after reviewing the predicted future outcomes.

Data Collection (S3–S4). We collect reflective annotations and temporal alignment data through the following stages:

- S3 (Vision-Reviewing, 27K): To fully utilize visual imagination, we introduce a reflective reviewing stage by generating paired trajectories from identical initial states:
 - Negative trajectory: Generated by the previously trained RIG-basic model, yielding suboptimal outcomes X^-, Y^-, A^- .
 - Positive trajectory: Generated by the superiorperforming policy STEVE-1 [20], yielding optimal outcomes X^+, A^+ .

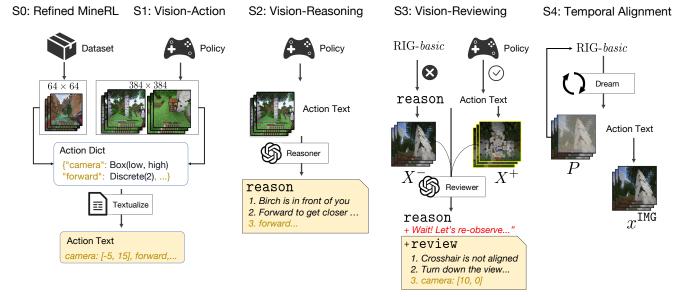


Figure 2. **Illustration of the data collection pipeline (S0–S4).** Note that at S3 (Vision-Reviewing), we run the trained RIG-basic and policy model (STEVE-1 [20]) in parallel, keeping instances where RIG-basic performs poorly compared to STEVE-1.

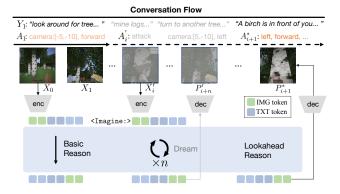


Figure 3. **Inference process in RIG.** RIG follows a structured *conversation flow* through multi-turn interactions. It consistently uses the fixed word Imagine: to clearly separate internally imagined scenarios from real observations, thereby guiding coherent reasoning, action prediction, and visual imagination.

We then adopt GPT-4o as a **Reviewer** to explicitly compare these parallel trajectories and generate refined reasoning: $Y^+ = \mathbf{Reviewer}(X^-, Y^-, A^-, A^+)$, so that we get corrective reasoning annotations:

$$Y = \{Y^-, \text{``Wait! Let's re-observe...''}, Y^+\}.$$
 (4)

This reflection annotation significantly enhances the ability of the model to review and correct reasoning mistakes.

• S4 (Temporal Alignment, 38K): We further generate multi-step imagined visual predictions (P) and explicitly align them with observed ground-truth visual tokens (x^{IMG}) to enhance long-horizon stability: $P_{i+1} \rightarrow x_{i+1}^{\text{IMG}}$.

Lookahead Reasoning with Imagination. Training on

datasets from stages 3 and 4 produces RIG-lookahead, a model that performs reasoning conditioned on imagined futures. stage 3 adopts Rejection Sampling Fine-tuning (RFT) to improve reasoning through model-generated rollouts. We apply RFT in embodied agents by leveraging joint reasoning and visual generation, which enables self-prediction of future states, previously infeasible due to the lack of visual prediction. Only the positive trajectory Y^+ is optimized, while the negative Y^- is excluded from loss, encouraging better self-correction.

RIG-lookahead simulates "dream trajectories" before acting. As shown in Fig. 3, imagined steps are marked with a fixed token "<Imagine:>" to distinguish from observations, allowing decisions to be refined by looking n steps ahead:

$$(Y_{i+1}^*, A_{i+1}^*, P_{i+1}^*) \stackrel{\mathcal{F}}{\leftarrow} (X_i, P_{i+1}, Y_{i+1}, ..., P_{i+n}, Y_{i+n}).$$
 (5)

This lookahead mechanism enables internal review and correction, reducing trial-and-error interactions and enhancing decision robustness in complex embodied tasks, as shown in Fig. A3.

4. Experiments

We conduct comprehensive experiments to validate the effectiveness of RIG across diverse tasks, focusing on data efficiency, scalability, and the benefits of integrating generation, reasoning, and lookahead. Evaluations on embodied tasks are performed under both *Manual* (hand-only) and *Tool* (e.g., iron pickaxe) to assess performance in varied em-

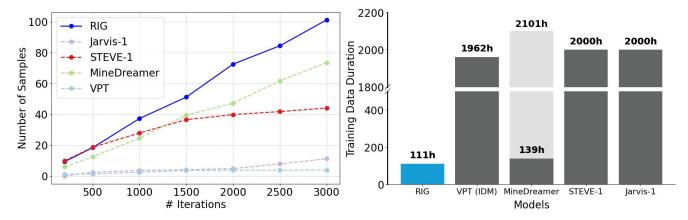


Figure 4. **Performance and data-efficiency comparison.** RIG-*basic* significantly outperforms other baselines with higher sample efficiency and achieves superior performance using only **111 hours** of training data (42h S0 MineRL-V0 and 69h S1-S4). MineDreamer [48], a hybrid-system model, separately trains a visual generation model (139 hours) but also relies on VPT for the policy model, increasing total data requirements. Duration of VPT [2] reflects only the IDM data used, measured as video frames, while STEVE-1 [20] and Jarvis-1 [35] also leverage the VPT dataset.

bodied scenarios.

4.1. Inplementation Details

RIG is initialized from the pretrained Janus-1.4B [8] with a sequence length of 4096 tokens. It operates as a fully endto-end agent, integrating textual reasoning, visual understanding, planning, and self-review within a unified Transformer model. For visual understanding, we use SigLIP-Large-Patch16-384 [41], while for visual generation RIG employs a VQ-based encoder with a 16,384-codebook and a 16× downsampling factor, each with two-layer MLP adaptors. A VQ tokenizer [31] converts images into discrete IDs, which are embedded and concatenated with text for multimodal processing. The training utilizes sequence packing and mixed data types, conducted on XTuner-lite [10]. The experiments on embodied tasks mainly follow the setup of STEVE-1 [20], evaluated by the number of samples for collection tasks and accuracy for exploration, both derived from environment feedback.

4.2. Main Results

We first evaluate the data efficiency of RIG against prior approaches (VPT [2], STEVE-1 [20], Jarvis-1 [35], and Mine-Dreamer [48]). We then benchmark its performance across three core task categories: Embodied Tasks, Generation Tasks, and Understanding & Reasoning Tasks. For more qualitative analysis, please refer to the case study in Fig. A5.

Data Efficiency and Training Duration. As shown in Fig. 4, RIG significantly surpasses other methods in terms of collected samples per iteration while requiring drastically lower total training time. Notably, RIG achieves superior performance using only 111 hours of total data collection, considerably less than other baselines (VPT: 1962h, Mine-

Dreamer: 2101h (139h + VPT), STEVE-1 and Jarvis-1: nearly 2000h).

Performance in Embodied Tasks. As shown in Fig. 4, RIG-basic have surpassed all other baselines with **93.4%** accuracy and **101.1** collected samples. RIG-lookahead (extended from RIG-basic) produce even greater progress than RIG-basic, with the highest accuracy of **94.1%** and **246.6 collected samples**. For extended analysis, we conduct additional scalability and ablation studies, as shown in Fig. 6 and Tab. 1.

Performance in Generation Tasks. As shown in Fig. 5, RIG-lookahead achieves the best generation quality among all baselines, with the lowest FID (77.6) and highest PSNR (18.4) compared to MineDreamer and Oasis. As shown in Tab. 2, performance consistently improves as capabilities are progressively integrated, with the most substantial gains attributed to the inclusion of lookahead reasoning. RIGlookahead significantly enhances generation performance over variants without it (e.g., ID3, FID: 156.5, PSNR: 17.9). Performance in Understanding & Reasoning. As shown in Fig. 5, RIG-lookahead demonstrates superior reasoning capabilities, achieving a Reasoning score of 7.3, surpassing GPT-40 (5.5), and an Understanding score of **9.6**, on par with GPT-4o (9.7). Additionally, Tab. 2 further validates the benefit of synergizing components, as reasoning and reviewing capabilities jointly contribute to stronger task understanding.

4.3. Analysis of Scalability

We evaluate the scalability of RIG along three key dimensions, training data ratio, iteration count, and inference steps, as illustrated in Fig. 6.

Training Scalability. RIG exhibits strong scalability with

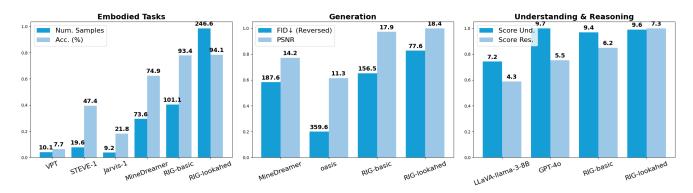


Figure 5. Comparison with various baselines across embodied tasks, generation, understanding, and reasoning. RIG-basic incorporates reasoning without reviewing, while RIG-lookahead integrates both reasoning and reviewing capabilities.

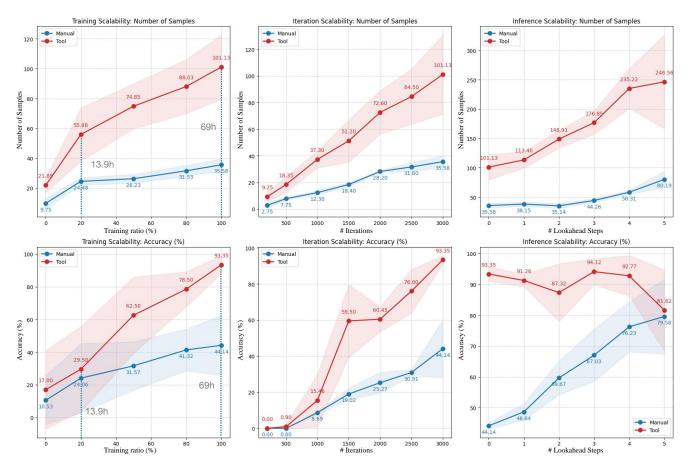


Figure 6. **Scalability Evaluation Across Training, Iteration, and Inference.** We evaluate the scalability of RIG by testing its performance across three different aspects: **training scalability, iteration scalability**, and **inference scalability**. Each column corresponds to a different scalability setting. The top row presents the number of collected samples in material-gathering tasks, while the bottom row reports the success rate in exploration-based tasks. Shaded regions represent variance. We exclude 42h MineRL-V0 pretraining from the total 111h in Figure. The training ratio is only counted before the lookahead reasoning.

training data volume. Increasing training data from 10% to 100% dramatically improves performance, especially at the 20% training threshold, where accuracy jumps substantially (manual: $10.53\% \rightarrow 24.06\%$, tool: $17.0\% \rightarrow 62.50\%$).

Data diversity significantly increases at this point, allowing the agent to encounter and adapt to a broader spectrum of complex scenarios. Beyond 20% data usage, the rate of accuracy improvement stabilizes, indicating the train-

ID		Ca	apabilities		Number of Samples					Accuracy (%)				
	Action	Gen.	Reason	Lookahead	wood	grass	dirt	avg.	Δ	Dig	Explore	Tower	avg.	Δ
Manual (ID 0–4)														
0	✓				7.9	6.2	8.9	7.7	+0.0	9.1	11.7	4.4	8.4	+0.0
1	✓	✓			11.0	16.5	12.1	13.2	+5.5	12.2	36.8	41.8	30.3	+21.9
2	1		✓		17.3	24.5	22.5	21.4	+13.8	34.2	31.8	37.8	34.6	+26.2
3	✓	✓	✓		22.2	45.9	38.7	35.6	+27.9	29.2	65.2	37.9	44.1	+35.7
4	1	✓	✓	✓	28.3	137.5	74.8	80.2	+72.5	65.8	84.2	88.7	79.6	+71.2
Tool	! (ID 0–4)													
0	/				24.6	33.1	42.4	33.4	+0.0	17.9	11.7	8.2	12.6	+0.0
1	✓	✓			25.8	29.9	48.4	34.7	+1.3	27.2	36.8	41.8	35.3	+22.7
2	✓		✓		26.9	49.9	51.0	42.6	+9.2	24.2	31.8	29.8	28.6	+16.0
3	1	1	✓		79.4	115.3	108.6	101.1	+67.7	85.1	100.4	94.7	93.4	+80.8
4	✓	✓	✓	✓	128.7	295.6	315.5	246.6	+213.2	95.4	84.2	102.8	94.1	+81.5

Table 1. **Ablation study on embodied tasks under different capability settings.** We compare different combinations of Action, Generation (Gen.), Reasoning (Reason), and Reviewing (Review). The table is divided into two groups: *Manual* (ID 0–4) and *Tool* (ID 0-4). The "Num." column represents the number of completed collecting tasks (wood, grass, dirt), while "Acc." denotes the success rate of exploration tasks. The columns "avg." is the average performance. For both metrics, we report the absolute values, along with the improvement (+x) over the baseline (ID 0 for *Manual* and *Tool*).

ID		Ca	apabilities		Gene	eration	Unders	Reasoning	
	Action	Gen.	Reason	Lookahead	$FID\downarrow$	PSNR ↑	Score-Stc.↑	Score-Env.↑	Score-Env.
0		✓			214.5	16.4	-	-	-
1	1	✓			225.6	16.3	-	-	-
2	/		✓		-	-	9.0	7.8	6.1
3	1	✓	✓		156.5	17.9	9.4	8.4	7.3
4	/	1	1	✓	77.6	18.4	9.6	8.1	8.5

Table 2. **Ablation study on Generation, Understanding, and Reasoning performance.** We compare different combinations of Action, Generation (Gen.), Reasoning (Reason), and Lookahead capabilities. "Score-Env." represents the environment-specific evaluation score from online understanding testing, while "Score-Env." denotes reasoning-specific evaluation. "Score-Stc." is computed on the static dataset STEVE-21K [43], and "FID" / "PSNR" measure image generation quality.

ing paradigm reaches a steady state and that the agent's action diversity nears its upper bound. With full training data (100%), RIG achieves superior results, outperforming existing approaches such as VPT [2] and Jarvis-1 [35] in accuracy and even surpassing STEVE-1 [20] in task collection efficiency, reaching 101.13 collected tasks and 93.35% accuracy. Notably, these results are attained purely through forward reasoning, without lookahead, suggesting substantial untapped potential for further enhancement by incorporating advanced reasoning mechanisms.

Iteration Scalability. RIG demonstrates robust performance growth over iterations. Under standard forward inference, task collection grows consistently from 9.25 samples at iteration 200 to 101.13 at iteration 3000, particularly pronounced in tool-assisted tasks, showing rapid convergence due to effective data utilization and stable trajectory

patterns. However, variance, illustrated by shaded areas, tends to increase with iterations, reflecting longer and more diverse trajectories that introduce complexity and fluctuation. Tasks involving exploration, which inherently contain more combinatorial subtasks (*e.g.*, material gathering followed by building structures), show larger variance and complexity over time. It potentially highlights the model's adaptive response to increasingly diverse scenarios.

Inference Scalability. The results from different lookahead steps demonstrate significant benefits from lookahead reasoning. Evaluating from the baseline at 3000 iterations, increasing steps (generating "dream trajectories") substantially improves performance. Task collection metrics exhibit rapid initial improvement and relatively low variance up to four steps, indicating accurate and stable trajectory predictions. However, variance increases at five steps, sug-

gesting accumulated prediction errors or hallucinations become more prominent. For accuracy metrics, tool-assisted tasks maintain high performance (peaking at 94.12% at 3 steps), with a slight decrease afterward due to ceiling effects and increased prediction uncertainty. Conversely, manual tasks show consistent performance improvement through stepwise lookahead, significantly benefiting from iterative reasoning, reaching a peak of 79.58% accuracy at 5 steps.

4.4. Ablation Study on Embodied Tasks

Effect of Generation. Incorporating visual generation significantly boosts performance, especially in exploration-based tasks. Comparing ID 0 and ID 1, the addition of visual generation yields clear gains in resource collection (*Number of Samples*, +5.5) and exploration accuracy (*Acc.*, +21.9) under the Manual setting, with corresponding improvements of +1.3 and +22.7 under the Tool setting. Visual generation improves the model's ability to interpret action outcomes, particularly in tasks requiring precise targeting. For example, misalignment of the crosshair in mining tasks can drastically affect success. Generating future visual states enables better action-state alignment, leading to more informed decisions.

Effect of Reasoning. Incorporating explicit reasoning yields substantial improvements across both Manual and Tool settings. Compared to ID 0, ID 3 with reasoning increases the Number of Samples from 7.7 to 35.6 (+27.9) and accuracy by +35.7 in the Manual setting; in the Tool setting, it boosts sample count by +67.7 and accuracy by +80.8. These gains arise from two key effects: (1) reducing redundant actions, without reasoning, autoregressive policies tend to overuse actions like "attack"; and (2) enabling more strategic, goal-directed exploration, reasoning helps the model better interpret its environment, especially in multi-step tasks like *Tower*, where resource gathering and construction must be planned jointly. When combined with generation (ID 2), even earlier variants already show consistent performance gains (e.g., Manual: +13.8 samples, +21.9 accuracy), highlighting the synergistic value of reasoning.

Effect of Lookahead Reasoning. Introducing reviewing brings the most significant gains. In the Manual setting, ID 4 achieves the best performance: *Number of Samples* rises from 7.7 to 80.2 (+72.5), and accuracy reaches 79.6% (+71.2). In the Tool setting, improvements are even more pronounced: from 33.4 to 246.6 (+213.2) in sample collection and from 12.6% to 94.1% (+81.5) in accuracy. As illustrated in Fig. 3, self-reviewing enables the agent to anticipate and reason over imagined future states, refining actions before execution. Notably, this capability only requires a small amount of additional data, 27K samples (0.8 hours), compared to thousands of hours used by other methods, demonstrating high efficiency.

Effect of Synergizing. As shown in Tab. 1, the synergy of

Action, Generation, Basic Reasoning and Lookahead Reasoning leads to the most robust performance, which enables structured learning and improves short-term decisions and long-horizon task completion. In the Manual setting (ID 4), there are significant gains in sample collection (from 7.7 to 80.2, +72.5) and accuracy (from 8.4% to 79.6%, +71.2). In the Tool setting, the impact is even greater, with sample counts rising from 33.4 to 246.6 (+213.2) and accuracy from 12.6% to 94.1% (+81.5), highlighting the effectiveness of the unified framework.

4.5. Ablation Study on Generation Quality

Effect of Generation. As shown in Tab. 2, visual generation significantly impacts overall system performance. Comparing ID 0 and ID 1, introducing action capability slightly degrades FID performance (from 214.5 to 225.6). This suggests that simply incorporating action learning without proper synergy with reasoning may introduce inconsistencies in the learned visual representations. However, when both action and reasoning are enabled (ID 3 and ID 4), generation quality improves substantially, with FID dropping to 156.5 and PSNR rising to 17.9.

Effect of Reasoning. Adding reasoning capabilities (ID 2 and ID 3) improves understanding and environmental interaction. Comparing ID 2 with ID 3, we see an increase in *Score-Static* (9.0 to 9.4) and *Score-Env.* (6.1 to 7.3). This demonstrates that reasoning enhances decision-making and allows for more informed perception of environmental dynamics. When reasoning is combined with generation (ID 3), RIG benefits from enhanced action planning and prediction, further improving overall performance.

Effect of Lookahead Reasoning. Including lookahead reasoning (ID 4) results in the best performance across all evaluation metrics. FID improves significantly (from 156.5 to 77.6), PSNR reaches 18.4, and *Score-Static* and *Score-Env.* achieve their highest values at 9.6 and 8.5, respectively. Lookahead reasoning enhances visual prediction by improving reasoning about future visual states.

Effect of Synergizing Reasoning and Imagination. By jointly optimizing generation, reasoning, and reviewing, RIG achieves the best trade-off between action prediction, visual understanding, and environmental reasoning. The best model (ID 4) shows that lookahead reasoning enhances decision-making, improving sample efficiency and interaction robustness. This underscores the benefit of integrating multiple modalities for coherent perception and action. As shown in Tab. 2, enabling lookahead reasoning substantially enhances image-generation quality. RIG-lookahead (ID 4) attains the lowest FID (77.6) and highest PSNR (18.4), significantly surpassing variants without lookahead reasoning (e.g., ID 3, FID: 156.5, PSNR: 17.9).

5. Conclusion

This paper introduces RIG, an end-to-end Generalist policy that integrates Reasoning and Imagination with superior adaptability and robustness in open-world environments. RIG unifies the understanding and generation of visual generation, action, and textual reasoning within a single autoregressive Transformer, and is capable of reasoning and planning by looking ahead with the dreamed trajectories to further improve its robustness. RIG obtains new state-of-the-art performance across embodied tasks, image generation, and reasoning tasks, with higher sample efficiency and generalization. RIG also exhibits higher scalability with training and test-time compute. We hope the results of RIG could inspire future research on synergizing reasoning and imagination in embodied agents.

A. Appendix

The appendix is organized as follows:

- Inference Pipeline (Appendix A.1) illustrates how RIG
 performs end-to-end inference: generating textual reasoning, imagined visual rollouts, and executable actions, with
 explicit use of the <Imagine:> token to support selfreview and temporal consistency.
- Training Pipeline (Appendix A.2) presents our multistage training framework, including offline supervised learning, GPT-4o-based reasoning and review relabeling, and imagination-grounded alignment strategies that enable lookahead-based decision-making.
- Data Distribution (Appendix A.3) analyzes the diversity
 of embodied tasks within our datasets, and illustrates how
 data volume scales with task complexity—from atomic
 skills like collection to composite ones like exploration
 and construction.
- Component Comparison (Appendix A.4) offers a systematic comparison with existing models, emphasizing RIG's unique capabilities in multimodal alignment, action granularity, and unified policy formulation without relying on task-specific modules.
- Tokenizer and Base Model Selection (Appendix A.5) explains our design choice of combining LlamaGen's VQ tokenizer with Janus as the vision-language foundation, offering a lightweight and effective setup for image-text grounding in Minecraft-like settings.
- Qualitative Results and Case Study (Appendix A.6) showcases examples where RIG performs internal reasoning, detects failure cases via self-review, and corrects actions before execution. We further compare it to GPT-40, demonstrating that strong visual generation alone does not guarantee robust policy reasoning.
- Multi-Modal Understanding Evaluation (Appendix A.7) evaluates RIG's embodied knowledge across diverse functional categories using the STEVE-

- 21K QA benchmark, covering survival, crafting, entity understanding, and more.
- Multi-turn Visual Reasoning Format (Appendix A.8) details our multi-round reasoning and imagination format, which supports fine-grained learning of vision-language-action alignment through step-by-step trajectory prediction.
- Environment Details (Appendix A.9) describes our experimental platform based on MineRL [13], featuring low-level egocentric control and programmable environment setup for robust and reproducible embodied evaluation.

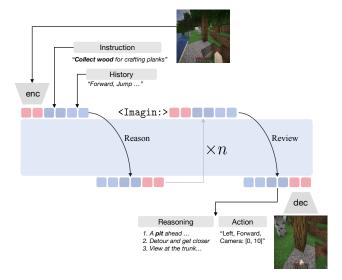


Figure A1. **Detailed inference pipeline.** RIG generates imagined visual states and corresponding reasoning to simulate multiple action trajectories, enabling self-review and corrective prediction.

A.1. Inference Pipeline

As illustrated in Fig. A1, RIG follows a multimodal autoregressive generation process. Given current observations and the task, the model produces (i) textual reasoning, (ii) low-level actions, and (iii) visual predictions of future frames. These imagined states, denoted by the fixed token <Imagine:>, are recursively fed back for internal reviewing and decision refinement. This mechanism allows iterative planning without environmental interaction.

A.2. Training Pipeline

As illustrated in Fig. A2, the training of RIG proceeds in four progressive stages:

• **S0/S1. Offline Supervised Fine-tuning (SFT):** The model learns to align the *dream flow* (model-generated predictions) with the *real flow* (observed data) through supervised learning. This phase improves visual state prediction quality, enhancing the accuracy of subsequent ac-

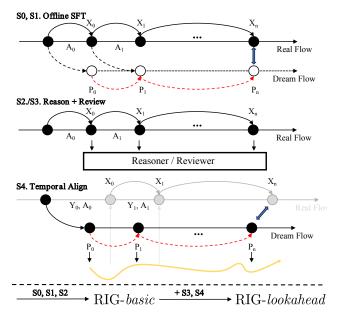


Figure A2. **Training pipeline of RIG.** S0/S1 pretrain the model by aligning real and imagined flows. S2/S3 enhance reasoning and reviewing via GPT-40 relabeling. S4 aligns temporally predicted trajectories (dream flow) with environment-grounded traces.

tion decisions.

Input: past frame, past action, task. *Output:* subtask, next action, next frame.

• S2. Reasoning Relabeling: A two-step process enhances decision quality. (1) An environment-based evaluator filters high-quality trajectories. (2) GPT-40 acts as a Reviewer to generate explicit reasoning traces and refined labels.

Input: past frame, past action, task. *Output:* reasoning, next action, optionally lookahead reasoning, next frame.

 S3. Review Relabeling: The trained model interacts in the environment, and an evaluator filters poor trajectories. GPT-40 as a Reviewer analyzes the imagined traces and relabels corrections for better trajectory quality.

Input: past frame, past action, task, imagined frame (<Imagine:>). *Output:* lookahead reasoning, corrected action, next frame.

• **S4. Temporal Alignment:** An imagined dream trajectory is generated via the autoregressive model. The entire sequence is behavior-cloned into the real environment, enabling frame-by-frame alignment and relabeling via the Reasoner.

Input: dream trace (states/actions). *Output:* real visual alignment, updated reasoning annotations.

Stages 0–2 are used to train RIG-basic, while Stages 3–4 further enhance RIG-lookahead with imagination-based alignment and long-horizon correction.

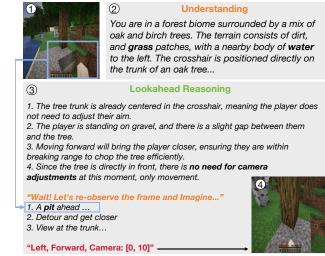


Figure A3. Qualitative example of lookahead and review. The agent understands the environment (1–2), simulates future states (3), and refines its decision through internal review before acting (4), successfully avoiding a hidden hazard.

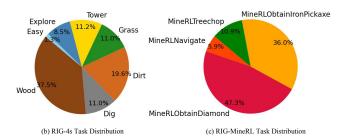


Figure A4. **Task distribution.** Our datasets include various embodied tasks with varying complexity, ensuring strong generalization across downstream goals.

A.3. Data Distribution

Fig. A4 visualizes the task distribution across our training datasets, which cover a spectrum of embodied scenarios such as resource collection, tower building, and exploration. As task complexity increases, we progressively expand the dataset size to ensure adequate supervision. Notably, harder tasks like building structures require significantly more data than simpler ones like gathering materials, highlighting the varying difficulty levels and skill composition in our training corpus.

A.4. Component Comparison

As summarized in Tab. A1 and Tab. A3, we compare RIG to prior works along multiple dimensions, including input modality, action granularity, and reasoning capabilities. Unlike prior methods relying on handcrafted API actions or curated codebooks, RIG operates solely on raw pixels and outputs keyboard-mouse controls, offering higher flexibil-

Method	Vision Encoder	Parameters	Vision Quality (Gen.)	MM Quality (Und.)	Evaluations
Autoregressive (A	AR)				
Emu3 [34]	VQ (D)	8B	0.68	-0.1	POPE, SEEDBench-Img, VQAv2 (85.2, 68.2, 75.1)
LlamaGen [31]	VQ (D)	111M, 343M, 775M, 1.4B, 3B	0.68	-0.34	-
Chameleon [22]	VQ (D)	7B, 34B	0.68	-0.29	VQAv2 (69.6)
Anole [9]	VQ (D)	7B	-	-	-
Janus [8]	VQ (D)	1.3B	0.68	-0.07	POPE, VQAv2 (87, 77.3)
AR + Diffusion					
Show-o [38]	Magvitv2 (D/C), Clip-ViT (C)	1.3B	0.68	-0.15	POPE, VQAv2 (84.5, 74.7)
Transfusion [47]	VAE (C)	0.16B, 0.37B, 0.76B, 1.4B, 7B	0.68	-0.01	-
Fluid [12]	VQ (D), VAE (C)	369M, 665M, 1.1B, 3.1B, 10.5B	0.68	0.02	-

Table A1. **Comparison of various unified multimodal methods**, categorized by their training approach (Autoregressive and AR + Diffusion), detailing vision encoder type, parameter scale, vision generation quality (GenEval SD3 8B), multimodal understanding quality, and evaluation benchmarks.

Tokenizer	Compression Ratio	Quantization	MS-COCO				ImageNet-1K			FFHQ			CelebA-HQ		
Tokemilei			PSNR↑	SSIM↑	rFID↓	PSNR (Minecraft)↑	PSNR↑	SSIM↑	rFID↓	PSNR↑	SSIM↑	rFID↓	PSNR↑	SSIM↑	rFID↓
Open-MAGVIT2 [23]	16 × 16	LFQ	30.06	0.502	6.649	27.21	29.62	0.398	2.701	31.77	0.774	1.994	32.36	0.844	2.865
LlamaGen [31]	8 × 8	VQ	30.71	0.616	4.123	28.93	30.28	0.498	1.403	33.39	0.868	0.701	34.82	0.937	0.502
LlamaGen [31]	16×16	VQ	29.93	0.491	6.077	27.06	29.81	0.448	1.657	31.58	0.772	1.366	32.18	0.837	1.113
Cosmos-Tokenizer-DI [1]	8 × 8	FSQ	31.74	0.730	4.564	30.84	31.73	0.725	1.841	35.35	0.892	0.555	37.77	0.948	0.261
Cosmos-Tokenizer-DI [1]	16×16	FSQ	30.74	0.591	12.252	29.91	30.69	0.582	6.529	33.17	0.808	7.663	33.86	0.854	5.953
Emu-3 [34]	16×16	VQ	-	-	-	24.16	-	-	-	-	-	-	-	-	-

Table A2. Comparison of Tokenizers across different benchmarks. PSNR, SSIM, and rFID are measured on MS-COCO, ImageNet-1K, FFHQ, and CelebA-HQ datasets. PSNR for Minecraft images is provided separately.

Method	VPT [2]	DreamerV3 [16]	DECKARD [26]	DEPS [36]	Plan4MC [40]	Voyager [33]	STEVE [20]	RIG (Ours)
Demos	Videos	None	Videos	None	None	None	Videos	Videos
Rewards	Sparse	Dense	Sparse	None	Dense	None	None	None
Observations	Pixels Only	Pixels & Meta	Pixels & Inventory	Feedback & Inventory	Pixels & Meta	Feedback & Meta & Inventory	Pixels & Feedback & Meta & Inventory	Pixels Only
Actions	Keyboard & Mouse	Discrete	Keyboard & Mouse	Keyboard & Mouse	Discrete	Code	Code	Keyboard & Mouse
Reasoning				✓		✓	 ✓	✓
Generation								✓
Extra Database					9	172	210	-

Table A3. Comparison between RIG (Ours), and existing works. This system-level comparison of LLM-based and RL-based methods focuses on data sources, reward setup, observation type, action representation, iterative planning, and skill database usage.

ity and lower task bias. Notably, our design unifies reasoning and generation into a single transformer policy with self-review and imagination steps, offering better trajectory-level coherence and enabling multi-turn lookahead.

A.5. Tokenizer and Base Model Selection

We adopt LlamaGen 16×16 VQ tokenizer and Janus-1.4B as our vision and language backbone. Tab. A2 reports their favorable reconstruction quality (PSNR 27.06) and semantic alignment. Janus uses a dual loss combining RGB and SigLIP-guided feature reconstruction, while LlamaGen provides discrete, compression-friendly tokens. Together, they form a scalable pipeline for visual imagination and reasoning, trained with simple cross-entropy objectives.

A.6. Qualitative Results and Case Study

Fig. A3 demonstrates the full inference cycle of RIG, where the agent understands the scene, reasons about its next move, simulates imagined outcomes, and conducts self-review before taking real action. In this wood-chopping task, the agent first identifies a tree in front, then reasons that moving forward seems viable. However, by simulating future states, it spots a hidden pit and triggers a self-correction: "Wait! Let's re-observe...". It updates its decision to Left, Forward, Camera: [0, 10]: right, successfully avoiding the hazard. This highlights our agent's ability to perform proactive planning, visual forecasting, and risk-aware correction through imagination and reviewing.

Fig. A5 further compares RIG-lookahead with GPT-40

image generation updated version. Both receive similar prompt and visual input. While GPT-40 generates a visually plausible prediction, it incorrectly judges the distance to the tree, prematurely issuing an attack command that leads to a deadlock. It continues to hallucinate progress without correcting the faulty assumption. In contrast, RIG accurately detects that the tree is blocked and unreachable, reasons about terrain features, and adjusts its position before action. The generated image aligns with the actual environment response, showing stronger spatial consistency and robustness in long-horizon decision-making.

A.7. Multi-Modal Understanding Evaluation

We further evaluate RIG on the STEVE-21K [46] benchmark, testing its general world knowledge and Minecraft-specific understanding. Drawing from the Minecraft Wiki and Reddit corpus, the dataset spans multiple knowledge dimensions:

- World Understanding: Terrain, entities, and biome behaviors.
- Player Mechanics: Combat, mobility, and health systems.
- Survival Strategies: Food sourcing, shelter, and threat avoidance.
- Resource Management: Gathering, mining, and inventory use.
- Crafting and Construction: Recipes and structural planning.
- **Tool Usage:** Equipment selection and upgrades.

We evaluate with 1000 QA pairs, categorized as: World & Entities (332), Mechanics & Survival (152), Knowledge & Discovery (108), Crafting (219), Tools (169), and Miscellaneous (20). Our model demonstrates strong accuracy and reasoning coherence across categories.

A.8. Multi-turn Visual Reasoning Format

To supervise step-level visual reasoning, we define a structured multi-turn dialogue format, as shown in below. Each entry logs the task instruction, prior action, current frame, reasoning, next action, and imagined future frame. This design aligns with autoregressive generation and supports fine-grained analysis and supervision.

- **Task Instruction:** Natural language goal (*e.g.*, "build a tower").
- **Previous Action:** Last executed action (*e.g.*, "camera:[0,10]").
- Current Frame: Visual observation from the environment.
- **Step Reasoning:** Textual reasoning for the next decision.
- Next Action: Predicted action.
- Next Frame: Imagined visual result of the action.

A.9. Environment Details

We use Minecraft as the testbed for embodied agents due to its open-ended nature and support for low-level human-like interactions. Agents act through egocentric RGB images and execute actions using keyboard and mouse inputs, making the environment ideal for sequential decision-making.

Our experiments are based on MineRL [13] v1.0 (Minecraft 1.16.5), which provides agents with first-person RGB observations and removes access to any privileged information. This version aligns with setups in prior works such as VPT [3] and STEVE-1 [20]. Agents only perceive visual inputs and interact through low-level actions, resembling human play.

A.10. Observation and Action Space

The agent receives 640×360 RGB images rendered from a first-person view with a 70-degree field of view. When the inventory is opened, the GUI and mouse cursor are visible. No voxel, depth, or structured APIs (e.g., "craft", "smelt") are used.

As shown in Tab. A3, the action space includes 14 types of keyboard and mouse operations, covering movement (WASD), item use, inventory management, and camera rotation (yaw, pitch). These mirror human inputs and do not rely on high-level abstractions.

For the camera action, which is originally a 2D continuous vector [a,b] representing pitch and yaw in $[-180^\circ,180^\circ]$, we quantize it into discrete steps of 5 degrees to adapt to language models, where $a{<}0/a{>}0$ denote looking up/down and $b{<}0/b{>}0$ denote turning left/right; all other actions are binary (0/1), indicating whether to execute them.

A.11. Environment Settings and Rules

To ensure diversity and generalization, each episode is initialized with a random agent position and world seed, exposing the model to varying terrains, structures, and objectives.

We adapt the MineRL [13] environment to support instruction-based initialization via a custom chat action, enabling programmatic task injection and faster evaluation cycles.

Programmatic Evaluation. For instruction-following tasks, we fix the seed and biome across different models to ensure a fair comparison. Each experiment runs under conditions most favorable for the given instruction, and multiple seeds are used across different trials.

Long-Horizon Tasks with Command Switching. In these scenarios, agents operate in a shared environment initialized with a common seed and biome. To reduce noise and sta-

Current IMG

Prompt



You are now a Minecraft simulator. I will give you the current frame and the task you need to do. You need to help me predict the best action and the next frame of the action. Action space {forward, back, left, right, jump, attack, sprint, ..., camera:[left or right, up or down]}.

Correct
Wrong

Current frame: , current task: **chop a tree**.

Predict: next action:..., next frame:

Reasoning Gen IMG Next IMG



RIG-lookahead

Reasoning:

- 1. In the first frame, the tree trunk in front of you is blocked by leaves but it is not close enough cannot attack immediately.
- 2 The tree on the right is closer and more suitable for immediate treecutting.
- 3. From the terrain point of view, you can attack the tree on flat ground by slightly adjusting to the right.

Next action: forward, sprint, camera:[0,10]







GPT-40 Image Generation Updated Version

According to the current screen, you are facing a tree, very close to the target, and the task is to "chop a tree". The best strategy should be to get close to the trunk and start attacking.

Prediction:

- Next action: attack (attack the trunk with your hands)
 Next frame speculation:
- Cracks will appear on the trunk (animation effect of attacking wood in Minecraft)
- Player position remains unchanged
- · HUD state is the same (no tools, only bare hands)
- · View is still aimed at the trunk





Figure A5. Case study comparison with GPT-4o. Given the same input and prompt (chop a tree), RIG reasons and imagines future states to choose a reachable tree and adjust position before acting. GPT-4o, despite high visual quality, misjudges the distance, executes an invalid action, and fails to revise its plan.

bilize training/testing, we apply the following environment rules:

- /difficulty peaceful: Disables hostile mobs.
- /gamerule doDaylightCycle false: Locks the environment in perpetual daytime.
- /gamerule keepInventory true: Prevents item loss upon death.

These configurations help maintain consistency across runs while preserving the task's difficulty and realism. They also support efficient benchmarking of embodied reasoning and planning in long-horizon, open-ended environments.

Prompt Format for Minecraft Conversation Reasoning

SYSTEM:

You are assisting in generating step-by-step reasoning data for Minecraft embodied agents in complex tasks like "build a tower". Each step should contain (1) the reasoning process, (2) the predicted next action, and (3) the generated next image frame. We follow a multi-turn conversation structure with visual context.

Input Format:

- Task: Given current task like "build a tower"
- Previous Action: e.g., camera:[0,10]
- Current Frame: Image observation

Human Prompt (Input):

```
<image>
```

Please make reasoning of action and predict the next action based on the current frame, current task and previous action, and generate the next frame based on the current frame and the next action.

```
The current task: build a tower. The previous action: camera:[0,10].
```

RIG Output (Response):

The reasoning: In the current frame, the task is to build a tower, but no suitable material is directly targeted. The player may be exploring for logs or stone. The previous action moved the camera upward to scout surroundings...

Next action: left, sprint, and next frame: <image>

Special Cases: Lookahead and Imagination

- When the model performs internal reflection or imagines the future, use the special token: <Imagine:> to prefix the reasoning block.
- Example: <Imagine:> Reasoning forward n steps into future environment outcomes.

Fields to be included in JSON:

• name, id, action, images, conversations, subtask (optional)

Example JSON Structure:

```
"name": "build a tower seed203",
  "id": 10,
  "action": "left, sprint",
  "conversations": [
    {
      "from": "human",
      "value": "<image>\n <Imagine:> Please make reasoning of action...
      task: build a tower...",
      "images": ["..._10_current.png"]
    },
      "from": "RIG",
      "value": "The reasoning: ... Next action: left, sprint,
      sand next frame: <image>",
      "images": ["..._10_next.png"]
   }
 ]
                                       14
}
```

References

- [1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 11
- [2] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. arXiv preprint arXiv: Arxiv-2206.11795, 2022. 5, 7, 11
- [3] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. Advances in Neural Information Processing Systems, 35:24639–24654, 2022. 2,
- [4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In RSS, 2023. 3
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.
- [6] Shaofei Cai, Zihao Wang, Xiaojian Ma, Anji Liu, and Yitao Liang. Open-world multi-task control through goal-aware representation learning and adaptive horizon prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13734–13744, 2023. 2
- [7] Shaofei Cai, Bowei Zhang, Zihao Wang, Xiaojian Ma, Anji Liu, and Yitao Liang. Groot: Learning to follow instructions by watching gameplay videos. *arXiv preprint arXiv:2310.08235*, 2023. 2
- [8] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Januspro: Unified multimodal understanding and generation with data and model scaling. arXiv preprint arXiv:2501.17811, 2025. 5, 11
- [9] Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. arXiv preprint arXiv:2407.06135, 2024. 11
- [10] XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. https://github.com/InternLM/ xtuner, 2023. 5
- [11] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. Advances in Neural Information Processing Systems, 35: 18343–18362, 2022. 2
- [12] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image

- generative models with continuous tokens. arXiv preprint arXiv:2410.13863, 2024. 11
- [13] William H Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: a large-scale dataset of minecraft demonstrations. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 2442– 2448, 2019. 2, 3, 9, 12
- [14] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*, 2020. 2
- [15] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*, 2021. 2
- [16] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104, 2023. 1, 2, 3, 11
- [17] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *ICML*, 2022. 2
- [18] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. In *Ijcai*, pages 4246–4247, 2016. 2
- [19] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. In *ICLR*, 2020. 2
- [20] Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. *arXiv preprint arXiv:2306.00937*, 2023. 2, 3, 4, 5, 7, 11, 12
- [21] Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to model the world with language. 2023. 1, 2, 3
- [22] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. arXiv preprint arXiv:2304.09842, 2023. 2, 11
- [23] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. arXiv preprint arXiv:2409.04410, 2024. 11
- [24] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. In RSS, 2023. 2
- [25] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample efficient world models. In *ICLR*, 2023.
- [26] Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. Do embodied agents dream of pixelated sheep?: Embodied decision making using language guided world modelling. arXiv preprint, 2023. 11
- [27] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *NeurIPS*, 2015. 2

- [28] OpenAI. Gpt-4v(ision) system card. 2023. 3
- [29] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. arXiv preprint arXiv:2205.06175, 2022. 3
- [30] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. 2
- [31] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv* preprint arXiv:2406.06525, 2024. 5, 11
- [32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023. 2
- [33] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. 1, 2, 11
- [34] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 2, 11
- [35] Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, et al. Jarvis-1: Open-world multitask agents with memory-augmented multimodal language models. *arXiv preprint arXiv:2311.05997*, 2023. 1, 2, 5, 7
- [36] Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multitask agents. arXiv preprint arXiv:2302.01560, 2023. 2, 11
- [37] Jialong Wu, Haoyu Ma, Chaoyi Deng, and Mingsheng Long. Pre-training contextualized world models with in-the-wild videos for reinforcement learning. In *NeurIPS*, 2023. 2
- [38] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. arXiv preprint arXiv:2408.12528, 2024. 2,
- [39] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multimodal models: Pretraining and instruction tuning. *arXiv* preprint arXiv:2309.02591, 2(3):3, 2023. 2
- [40] Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks. arXiv preprint arXiv:2303.16563, 2023. 11
- [41] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training.

- In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11975–11986, 2023. 5
- [42] Chi Zhang, Penglin Cai, Yuhui Fu, Haoqi Yuan, and Zongqing Lu. Creative agents: Empowering agents with imagination for creative tasks. *arXiv preprint* arXiv:2312.02519, 2023. 1
- [43] Zhonghan Zhao, Wenhao Chai, Xuan Wang, Li Boyi, Shengyu Hao, Shidong Cao, Tian Ye, Jenq-Neng Hwang, and Gaoang Wang. See and think: Embodied agent in virtual environment. arXiv preprint arXiv:2311.15209, 2023. 1, 2,
- [44] Zhonghan Zhao, Kewei Chen, Dongxu Guo, Wenhao Chai, Tian Ye, Yanting Zhang, and Gaoang Wang. Hierarchical auto-organizing system for open-ended multi-agent navigation. *arXiv preprint arXiv:2403.08282*, 2024. 1
- [45] Zhonghan Zhao, Ke Ma, Wenhao Chai, Xuan Wang, Kewei Chen, Dongxu Guo, Yanting Zhang, Hongwei Wang, and Gaoang Wang. Do we really need a complex agent system? distill embodied agent into a single model. arXiv preprint arXiv:2404.04619, 2024.
- [46] Sipeng Zheng, Jiazheng Liu, Yicheng Feng, and Zongqing Lu. Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds. arXiv preprint arXiv:2310.13255, 2023. 12
- [47] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. arXiv preprint arXiv:2408.11039, 2024. 2, 11
- [48] Enshen Zhou, Yiran Qin, Zhenfei Yin, Yuzhou Huang, Ruimao Zhang, Lu Sheng, Yu Qiao, and Jing Shao. Minedreamer: Learning to follow instructions via chain-of-imagination for simulated-world control. *arXiv preprint* arXiv:2403.12037, 2024. 1, 2, 5