Similar: A Step-Wise, Multi-Dimensional Reward Model for Virtual Agent Learning and Reasoning

Bingchen Miao[1,2,*], Yang Wu[2,*], Minghe Gao[1,*], Qifan Yu[1], Wendong Bu[1,2], Wenqiao Zhang[1], Yunfei Li[2], Siliang Tang[1], Tat-Seng Chua[3], Juncheng Li[1†]

Zhejiang University [1], Ant Group [2], National University of Singapore[3]
{miaobingchen23, minghegao, yuqifan, wendongbu, wenqiaozhang, siliang, junchengli}@zju.edu.cn,
{wy306396, qixiu.lyf}@antgroup.com, dcscts@nus.edu.sg

Abstract

The development of Generalist Virtual Agents (GVAs) has shown significant promise in autonomous task execution. However, current training paradigms face critical limitations, including reliance on outcome supervision and labor-intensive human annotations. To address these challenges, we propose `Similar`, a **s**tep-w**i**se **m**ult**i**-dimensiona**l** gener**a**list **r**eward model, which offers fine-grained signals for agent training and can choose better actions for inference-time scaling. Specifically, we begin by systematically defining five dimensions for evaluating agent actions. Building on this framework, we design an MCTS-P algorithm to automatically collect and annotate step-wise, five-dimensional agent execution data. Using this data, we train `Similar` with our crafted Triple-M strategy. Furthermore, we introduce the first benchmark in the virtual agent domain for step-wise, multi-dimensional reward model training and evaluation, named ***SRM***. This benchmark consists of two components: ***SRMTrain***, which serves as the training set for `Similar`, and ***SRMEval***, a manually selected test set for evaluating the reward model. Experimental results demonstrate that `Similar`, through its step-wise, multi-dimensional assessment and synergistic gain, provides GVAs with effective intermediate signals during both training and inference-time scaling.

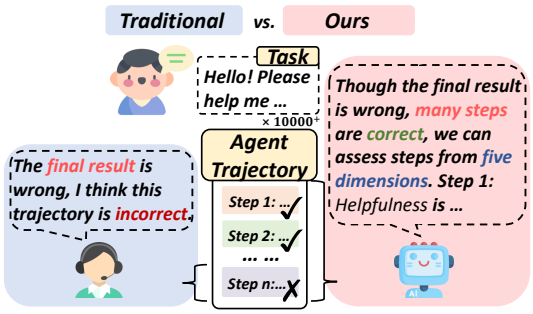
1 Introduction

Generalist Virtual Agents (GVAs) [10, 5] powered by Multimodal Large Language Models (MLLMs [18, 19, 23, 24]) process multimodal inputs (UI elements [48], text [28], visuals [41]) to navigate digital environments, performing tasks and generating outputs that manipulate interfaces or provide responses. The training of GVAs relies on outcome-based rewards from human-annotated trajectories, where task completion serves as the primary supervision signal [13].

However, this paradigm with the outcome reward for GVAs has significant limitations. **1) Lack of multi-dimensional fine-grained process supervision:** Existing methods typically focus on global task success or the final state of the task, overlooking intermediate steps in execution [42]. This oversight makes it impossible to pinpoint failures in unsuccessful trajectories or identify errors in successful ones, resulting in inefficient learning and reasoning processes [31, 21, 11]. In contrast, a Process Reward Model (PRM) offers a better alternative by providing fine-grained supervision signals to guide agent behavior. **2) Reliance on human-annotated trajectories with reward signals:** Domain experts need to meticulously annotate trajectories consisting of dozens of steps with accurate

[*]Equal contribution
[†]Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Scaling Environments for Agents (SEA).



(a)

Work	PB	AA	MD	Platforms	Size
WebArena	✗	✗	✗	Web	812
VisualWebArena	✗	✗	✗	Web	910
WorkArena	✗	✗	✗	Web, Windows	19.9k
Android World	✗	✗	✗	Android	116
WebVoyager	✗	✓	✗	Web	300
OSWorld	✗	✓	✗	Linux, Windows	369
GUI Odyssey	✗	✓	✗	Android	7.7K
Mobile-Agent	✓	✗	✗	Android	30
OmniACT	✓	✗	✗	Web, Windows	9.8K
SRM (Ours)	✓	✓	✓	**Web, Linux, Windows, Android**	**110k**

(b)

Figure 1: (a) Traditional coarse-grained outcome-based labor-intensive paradigm vs. Our fine-grained process-based autonomous paradigm. (b) Comparison of benchmark platforms. Previous works focused on virtual agent benchmarks, while ours is the first benchmark specifically for virtual agent reward models. The PB, AA, and MD in the list represent Process-based, Automatic Annotation, and Multi-Dimension, respectively.

outcome-based rewards to train GVAs [13]. Furthermore, obtaining step-wise fine-grained process-based rewards makes the process labor-intensive, time-consuming, and nearly infeasible at scale [9, 6]. **3) Difficulty in scaling inference-time**. Recent outstanding work has demonstrated that inference-time scaling can significantly enhance agent performance [8, 36]. However, relying on result-based training with extensive human annotation limits the ability to handle complex tasks [29, 46]. Therefore, our focus has shifted to breaking result-oriented manual annotation-dependent training methods through step-wise automatic reward model.

To address these challenges, we propose `Similar`, a **s**tep-**wi**se **m**ult**i**-dimensiona**l** gener**a**list **r**eward model. It provides fine-grained supervision signals for agent training and inference-time scaling, enabling automated, multi-faceted assessment without relying on labor-intensive human annotations. Specifically, **1)** we introduce a step-wise, multi-dimensional assessment system for GVA actions, **defining five key dimensions of process supervision signals**: *Helpfulness*, *Odds of Success*, *Efficiency*, *Task Relevance*, and *Coherence*. These dimensions are designed to minimize overlap while collectively providing a comprehensive assessment of each action's quality. **2)** Then we design an **MCTS-P** algorithm to **automatically collect and annotate tens of thousands of step-wise actions** based on the five dimensions. This approach is applied across four distinct environment domains: Web, Android, Linux, and Windows. Unlike existing methods that rely on labor-intensive human annotations, this automated framework ensures scalability across diverse environments and generates a unified, fine-grained dataset that captures universal reasoning patterns, significantly reducing the cost and time required for data collection. **3)** Finally, using this dataset, we employ a **Triple-M** (multi-step, multi-objective, and multi-modal) **strategy to train a reward model**. This strategy integrates multiple dimensions of assessment and generates a synergistic gain by combining the strengths of five dimensions. As illustrated in Figure 1 (a), traditional methods focus solely on outcomes, require significant manual effort, and are coarse-grained, outcome-based, and labor-intensive. In contrast, our approach enables `Similar` to perform step-wise, multi-dimensional automatic assessment of agent trajectories, making it fine-grained, process-based, and autonomous.

Since reward models are crucial for GVAs, and prior research has not focused on evaluating reward models, we propose ***SRM***, the first benchmark in the GVA domain for step-wise, multi-dimensional reward model training and evaluation. Figure 1 (b) illustrates that it consists of 110k automatically annotated data points, divided into the scalable ***SRMTrain*** (78k) for training `Similar` and the curated ***SRMEval*** (32k) for evaluating reward models.

Our reward model, `Similar`, can enhance the learning and reasoning of GVAs. **For training**, it serves as a reward model in a reinforcement learning framework, guiding GVAs to optimize its behavior based on action quality. By providing fine-grained feedback, it effectively guides the agents' learning process and enhances their performance. **For inference-time scaling**, it can be integrated with search algorithms such as Monte Carlo Tree Search (MCTS) to leverage reward signals for filtering candidate actions, and improve model performance [8, 45]. By selecting actions that are more likely to complete the task, it enhances accuracy and reduces time.

Extensive experiments demonstrate the superiority of our approach: **1)** Effectiveness of step-wise, multi-dimensional data: Using our collected data for reward modeling, `Similar`-RL-Llama achieves

a 13.2% improvement over the baseline Llama-3.2-11B-Vision model on the ***SRMEval*** benchmark, demonstrating the effectiveness of our automated framework in enabling fine-grained assessment of GVA actions. **2)** Synergistic gain from the Triple-M strategy: The Triple-M strategy integrates multiple dimensions by leveraging the strengths of five dimensions, enabling `Similar`-TM-Llama to achieve an Avg score of 61.2 on ***SRMEval***, significantly outperforming `Similar`-RL-Llama (53.9, a 13.5% improvement). This highlights the synergistic gain of our training strategy. **3)** Effective guidance in training and inference: `Similar` provides fine-grained, multi-dimensional feedback during training and integrates with search algorithms like MCTS to scale inference-time during inference to improve reasoning accuracy. Its strong performance across multiple benchmarks underscores its versatility and practical applicability.

Our contributions can be summarized as follows:

- We define five dimensions for step-wise GVA assessment and an MCTS-P algorithm to collect fine-grained, cross-platform reward model data annotations.
- We propose a Triple-M strategy to train a reward model, called `Similar`, integrating multiple dimensions and generating synergistic gains for robust, fine-grained feedback.
- Moreover, we introduce ***SRMEval***, a multi-step, multi-dimensional, and multi-platform benchmark for evaluating reward models, which is a set of ***SRM*** to advance research in reward model performance assessment.
- Experiments demonstrate that our approach, through step-wise multi-dimensional assessment, boosts GVAs during both training and inference-time scaling.

2 Related Work

2.1 Fine-Tuning Virtual Agent

Fine-tuning Virtual Agents traditionally relies on human-annotated datasets, which are labor-intensive and time-consuming [33]. Methods such as imitation learning [14] and reinforcement learning [3, 4] have been employed to fine-tune agents based on curated expert trajectories or outcome rewards, but these approaches often suffer from compounding errors and limited exploration [7, 38, 30]. Recent advancements, such as reject sampling fine-tuning (RFT) [44] and direct policy optimization (DPO) [25], have sought to reduce reliance on human annotations by leveraging both successful and failure trajectories [17, 50]. However, these methods face significant challenges, including the lack of process supervision and reliance on human-annotated data, which limit their scalability and adaptability [40, 13, 27]. In contrast, our work addresses these limitations by introducing a novel training paradigm that leverages multi-dimensional process supervision and automated annotation to enhance the learning and reasoning capabilities of GVAs.

2.2 Reward Models for Virtual Agent

Reward Models (RMs) are critical for guiding virtual agents by evaluating action quality [47, 51]. While Outcome Reward Models (ORMs) focus on task success [42, 43], Process Reward Models (PRMs) provide feedback on intermediate steps, offering an evaluation of agent performance in complex reasoning tasks [31]. Recent studies show that PRMs outperform ORMs in tasks like math reasoning, where process supervision is essential [21, 20]. However, generating high-quality process supervision data remains challenging, as human annotation is expensive. To address this, methods like step-level Q-value [47] and ReST-MCTS* [49] have explored MCTS to automate data collection, achieving significant gains. Building on these insights, our work introduces a step-wise, multi-dimensional system leveraging MCTS to collect fine-grained annotations, enabling a robust reward model to guide GVAs.

3 Method

In this section, we present the pipeline for training our `Similar` model. The ***SRM*** benchmark will be introduced in Section 4. As shown in Figure 2, to evaluate agent steps multi-dimensionally, we first define five-dimension process supervision (Section 3.1). Next, we introduce an MCTS-P algorithm to automatically collect step-wise annotations (Section 3.2). Finally, we design the Triple-M strategy to train `Similar`, achieving synergistic gains across five dimensions (Section 3.3).

3.1 Five-Dimensional Process Supervision Framework

To assess the quality of an agent's steps, we systematically define a five-dimensional process supervision framework. Given task complexity and interdependencies, a single metric is insufficient for assessing step quality [47]. Our framework addresses this limitation by covering the multi-faceted nature of step assessment. The first three dimensions—*Helpfulness*, *Odds of Success*, and

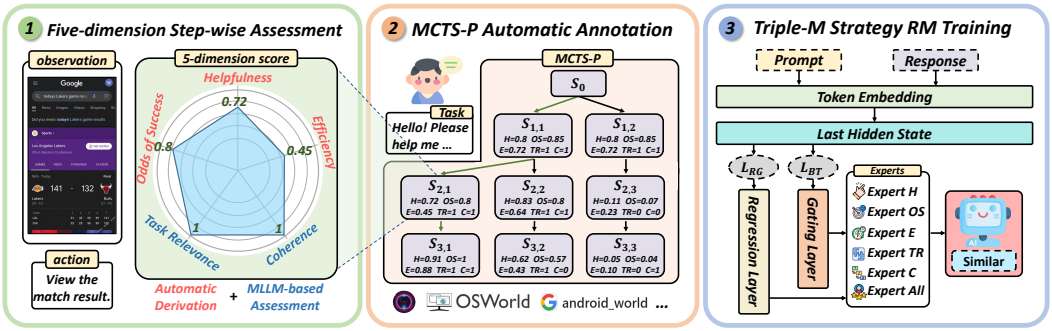


Figure 2: `Similar` **model training pipeline.** First, we systematically define five dimensions to describe the quality of an agent's step. Next, we propose an MCTS-P algorithm to automatically collect annotated step-wise data. Finally, we design the Triple-M strategy to train the `Similar` model, which can guide the agent during both the training and inference phases.

Efficiency—are computed automatically, while the remaining two—*Task Relevance* and *Coherence*—are assessed using MLLMs. These dimensions are independent and interpretable, ensuring broad applicability across tasks.

The current step is denoted as S_i, where i is the step index. The three automatic metrics are derived through MCTS simulations [22]. For S_i, we simulate N subsequent trajectories until a termination condition is met (i.e., the agent completes the task or reaches the maximum step length). We define the basic reward r_i as: $r_i = \begin{cases} 1, & \exists a_{i,j} \in A,\ a_{i,j} = a^* \\ 0, & \text{otherwise} \end{cases}$, $j \in N$, where a^* represents the ground truth, $a_{i,j}$ denotes the final action of the j-th trajectory in step i, and A is the set of all actions. The following sections detail how each dimension assesses S_i.

Helpfulness (H). It quantifies whether a given step contributes positively or negatively to task completion, assigning values inversely proportional to the trajectory length. This dimension is designed to assess the impact of each step on the overall task. Steps that facilitate task completion are considered helpful, while those that hinder progress are assigned negative values. For example, each step in a 3-step successful trajectory is worth $1/3$, while steps hindering progress (those failing to lead to success) receive corresponding negative values. And in two successful trajectories of the same task, the steps in the trajectory with fewer steps will have higher *Helpfulness* value.

The *Helpfulness* can be calculated as the following formula:

$$H_i = \frac{1 - AC_{i-1}}{M - i + 1}(2r_i - 1),$$

where $AC_i = \begin{cases} 0, & i = 0 \\ \max(AC_{i-1} + H_i, 0), & \text{otherwise} \end{cases}$, which is a mathematical placeholder to recursively track cumulative *Helpfulness* scores during MCTS rollouts. And M is the total number of reasoning steps.

Odds of Success (OS). It measures the probability that a given step will lead to the successful completion of the task. This dimension identifies steps that are more likely to result in a successful outcome. Steps with higher values are more likely to lead to success, while those with lower values are less likely to succeed. Conversely, incorrect steps lead to failure. The *Odds of Success* is calculated by evaluating the proportion of successful paths among simulated results from a given step.

The formula for *Odds of Success* is defined as:

$$OS_i = \frac{\sum_{j=1}^{N} \mathbb{I}(a_{i,j} = a^*)}{N},$$

where $\mathbb{I}(\cdot)$ is the indicator function.

Efficiency (E). It evaluates whether a given step is operationally efficient in terms of resource consumption, such as time or computational effort. A fundamental assumption is that fewer steps equate to higher efficiency, as shorter paths imply lower resource usage. Steps that reduce the total

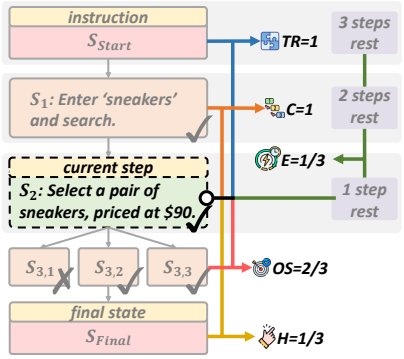


Figure 3: **An example describing the** 5 **dimensions**. Our 5 dimensions align with 5 task elements of an agent step: ***H*** (final state), ***OS*** (next step), ***E*** (number of steps), ***TR*** (instruction), and ***C*** (last step).

Algorithm 1 MCTS-P Algorithm

Input: Initial state s_0
Create root node S_0 with state s_0
while within computational budget **do**
  $S_i \leftarrow$ TreePolicy(S_0)
  $\Delta \leftarrow$ DefaultPolicy$(s(S_i))$ // Simulate random playout
  Backup(S_i, Δ) // Backpropagate
end while
return a(BestChild$(S_0, 0)$)
TreePolicy(S)
while S is nonterminal **do**
  if S not fully expanded **then**
    return Expand(S) // Expand tree
  else
    $S \leftarrow$ BestChild(S, C)
  end if
end while
return S
BestChild(S, c)
$v(S) = H(S) + OS(S) + E(S) + TR(S) + C(S)$
return $\arg\max_{S' \in \text{children of } S} \left(\frac{v(S')}{n(S')} + c\sqrt{\frac{2 \ln n(S)}{n(S')}} \right)$
Output: Action a

number of steps required to complete a task are considered efficient, as they enable the agent to accomplish the task more quickly and with fewer resources.

The *Efficiency* metric is calculated as the following formula:

$$E_i = \frac{Len_{i-1} - Len_i}{len_0},$$

where $Len_i = \text{avg}(Len(S_{i,j}))$, and $Len(S_{i,j})$ represents the number of steps remaining to complete the task after executing action $a_{i,j}$.

Task Relevance (TR). It assesses whether a step is related to the task instruction. Some steps may be task-relevant but still fail (e.g., recording incorrect notes), while others may be irrelevant yet contribute to success (e.g., clicking on a blank screen). These distinctions cannot be captured through automated calculations. However, MLLMs with advanced image understanding can evaluate this dimension. *Task Relevance* is binary, with values $\{0, 1\}$.

Coherence (C). It: measures the continuity and logical flow between consecutive steps. Some operations, although task-relevant, efficient, and likely to lead to success, may lack coherence with the previous step. For example, in a task such as "Query the Lakers' game result and record it in a Note," opening a browser and a Note simultaneously may lack coherence compared to directly searching for the game result after opening the browser. *Coherence* is also evaluated using MLLMs and is a binary classification dimension, with possible values of $\{0, 1\}$.

The prompts of two MLLM-based assessment dimensions is detailed in Appendix A. To better understand the five dimensions, we use Figure 3. Each agent step relates to five task elements: instruction, last step, next step, final state, and step count. In the figure, step S_2 is assessed by these dimensions. The task requires 3 steps: S_1, S_2, and $S_{3,2}$, with all sharing an *H* value of $\frac{1}{3}$. Among the next steps from S_2, $S_{3,2}$ and $S_{3,3}$ are correct, yielding an *OS* value of $\frac{2}{3}$. The *E* value is $\frac{1}{3}$, calculated as $\frac{2-1}{3}$. Since S_2 aligns with S_1 and the instruction, *TR* and *C* values are 1.

3.2 Automatic Generalist Dataset Collecting

MCTS (Monte Carlo Tree Search) is a heuristic search algorithm used in decision-making, combining random sampling and tree-based search to find the optimal option. Based on its advantages, such as its scalability and efficient exploration-exploitation balance [22, 35], we propose a modified version, MCTS-P, to automatically collect annotated data. MCTS-P leverages the five dimensions introduced in Section 3.1 to comprehensively assess each step taken by the virtual agent.

In MCTS-P, the five-dimensional scores are used as the basis for node selection and backpropagation. Specifically, the algorithm computes a weighted sum of the five dimensions to obtain a composite score for each step. This composite score serves as the value $v_{i,j}$ for each node $S_{i,j}$ in the search tree. The tree structure in MCTS-P is similar to traditional MCTS, with each node $S_{i,j}$ storing the action $a_{i,j}$, visit count $n_{i,j}$, and value $v_{i,j}$. The pseudo-code for MCTS-P is provided in Algorithm 1.

To build a comprehensive and generalist dataset for training and testing reward models, we collect a large number of task trajectories from agents across four different platforms: Web, Android, Linux, and Windows. Using the MCTS-P algorithm, we perform automatic data annotation to collect process supervision signals. The annotation process involves the following steps: **1)** For each node $S_{i,j}$ in the search tree T_q, we calculate the minimum number of steps M required to reach a correct answer. **2)** During the expansion phase, the algorithm simulates N possible outcomes for each step to obtain the basic reward r_i. **3)** Based on M and r_i, we compute the three automatically calculated dimensions: *Helpfulness*, *Odds of Success*, and *Efficiency*. **4)** We then use a MLLM (GPT-4o [15]) to evaluate the *Task Relevance* and *Coherence* of each step. **5)** We prune all incomplete branches (those that do not reach a final answer) and verify the correctness of the remaining paths using the evaluation methods provided by the four benchmark environments. The obtained trajectories are selected as the final dataset for training and evaluation.

3.3 Triple-M Strategy for RM Training

Traditional reward modeling relies on human-annotated data [32], whereas we generate step-wise annotations across multiple dimensions. To better utilize integrating Multi-step, Multi-dimensional, and Multi-modal data, we propose a novel **Triple-M strategy** tailored for virtual agents.

Our Triple-M strategy leverages a pre-trained decoder-only MLLM as the backbone feature extractor f_θ, divided into two stages. The first stage trains a regression layer for five-dimensional score prediction. For each input sequence $x \oplus y$ (where x represents the prompt and y represents the response), we extract the last hidden state $h \in \mathbb{R}^d$ with d-dimensional features and map it to a five-dimensional reward score through a linear regression layer $W \in \mathbb{R}^{d\times 5}$. The model is optimized using a regression loss:

$$L_{RG} = \min_{\theta,W} \mathbb{E}_{x,y,r\in\mathcal{D}} \|W^\top h - r\|_2^2,$$

where $r \in \mathbb{R}^5$ is the ground-truth reward vector, and $\mathcal{D}$ is the training dataset.

In the second stage, we train a gating network to dynamically balance the five-dimensional scores, addressing the multi-objective optimization problem. We introduce a prompt-aware gating network g_ϕ, implemented as a shallow multi-layer perceptron (MLP). This network dynamically adjusts the model's focus based on the input prompt x. The gating network computes non-negative coefficients $w \in \mathbb{R}^5$ for the five reward dimensions. These coefficients are derived from the last hidden state corresponding to the prompt x and normalized via a softmax function.

The gating network is trained using the Bradley-Terry (BT) loss [2] function, which aligns the model's predictions with human preferences. The BT loss is formulated as:

$$L_{BT} = \min_{\phi} \mathbb{E}\left[-\log \frac{\exp(R_{\text{chosen}})}{\exp(R_{\text{chosen}}) + \exp(R_{\text{rejected}})}\right],$$

where R_{chosen} and R_{rejected} represent the preference scores for the chosen and rejected responses. During training, only the gating network parameters are updated, while the backbone network and regression layer remain frozen.

Finally, the scalarized reward R of the trained `Similar` model is computed as $R = g_\phi(f_\theta(x))^\top r$. Through this design, our `Similar` model can not only output five-dimensional scores but also a comprehensive score that balances the five dimensions, just like an all-around expert combining the capabilities of six experts.

4 *SRM* Benchmark

We introduce the ***SRM*** benchmark, built from multi-dimensional, and multi-platform annotated data.

Data Collection. We used GPT-4o-1120 [15] as the agent to collect agent action trajectories across four benchmarks—WebArena (WA) [52], VisualWebArena (VWA) [16], Android World (AW) [26], and OSWorld (OW) [39]. Since these environments do not provide dedicated training and test sets, to ensure fairness and prevent data leakage, we rigorously used 70% data provided by these benchmarks for agent trajectory data collection, while the remaining 30% were reserved for evaluation experiments to ensure no data overlap between ***SRMTrain*** and evaluation sets. Ultimately, we collected $10k$ agent trajectories by generating multiple distinct actions per step through task-specific prompt injection and stochastic exploration. And we constructed $110k$ preference pairs for the ***SRM*** benchmark based on the scores of the designed dimensions. We also sampled some data for human experts to verify the accuracy of the pairs, as detailed in Appendix C.

Instruction	Observation	Step Idx	Trajectory	Type	Candidate Action Pair	
In Simple Calendar Pro, delete all the events.		*2*	*Step1: Click Search.*	*E*	*Click Calender.*	*Input text "Simple Calendar Pro".*
Add a exact product to my shopping cart.		*2*	*Step 1: Click menuitem "Grocery & Gourmet Food" menu.*	*H*	*Scroll down.*	*Click link "Pantry Staples"*

Figure 4: Data cases of ***SRMEval***.

Table 1: Performance comparison of common MLLMs, `Similar`-RL, and `Similar`-TM on ***SRMEval***.

Reward Model	H	OS	E	TR	C	Tot	Traj	Avg
GPT-4-Turbo	44.7	46.3	44.8	48.7	42.3	46.5	44.5	46.6
GPT-4o	49.9	50.1	47.9	51.4	43.8	51.1	49.8	51.4
InternVL-2.5	38.9	43.1	44.3	41.4	41.8	40.7	39.0	40.9
Qwen2-VL	45.7	42.1	41.7	44.5	42.1	43.2	41.6	42.9
+ `Similar`-RL	52.4	49.6	48.3	50.2	47.9	51.0	45.4	49.2
+ `Similar`-TM	**60.5**	**57.8**	**56.6**	**59.7**	**56.2**	**58.4**	**53.9**	**57.3**
Llama-3.2-V	48.2	47.6	47.1	51.1	42.6	49.5	44.5	47.6
+ `Similar`-RL	55.1	52.1	52.7	55.3	47.8	54.6	49.5	53.9
+ `Similar`-TM	**63.8**	**60.5**	**59.2**	**62.7**	**56.8**	**61.4**	**58.7**	**61.2**

Dataset Construction. We carefully selected 32k data points for manual annotation as the test set ***SRMEval***, while the remaining 78k data points were used as the training set ***SRMTrain*** to train the `Similar` model. The test data tasks are distinct from those in the training data. As shown in Figure 4, each data point in ***SRMEval*** includes instruction, observation screenshot, step index, trajectory, evaluation type, and candidate action pair. The evaluation types include our proposed five key dimensions—*Helpfulness* (H), *Odds of Success* (OS), *Efficiency* (E), *Task Relevance* (TR), and *Coherence* (C)—as well as a total dimension that integrates the five dimensions (Tot, weighted sum of the five dimensions) and a trajectory-level dimension (Traj, average Tot score of all steps in trajectory). More visualization cases of ***SRMEval*** are detailed in Appendix F.

New Task and Evaluation Metric. Based on ***SRMEval***, we proposed a new task for reward models in the virtual agent domain: *Selecting the better action from candidate action pair at step i in a specific dimension*. The evaluation metric is Accuracy, measuring the reward model's ability to select the better action. Accuracy is calculated under each evaluation type. For clarity, we use abbreviations such as H to represent each metric in our experiments.

5 Experiments

5.1 Experimental Setup

Baselines. We selected two baseline methods: **1)** Qwen2-VL-7B-Instruct [34] and Llama-3.2-11B-Vision-Instruct were directly used as reward models, with prompts provided (detailed in in Appendix A) to score agent steps. **2)** `Similar`-RL-Qwen and `Similar`-RL-Llama, whose backbones match the aforementioned models, were trained using reinforcement learning [1] on our ***SRMTrain*** dataset to score agent steps.

To benchmark against these baselines, we introduce `Similar`-TM-Qwen and `Similar`-TM-Llama, which are trained on the ***SRMTrain*** dataset using the Triple-M strategy with Qwen2-VL-7B-Instruct and Llama-3.2-11B-Vision-Instruct as backbones, respectively.

Evaluation Benchmarks. We first tested the preference alignment capability of `Similar` on our ***SRMEval***, compared with GPT-4o-1120, GPT-4-Turbo, and InternVL-2.5-8B. Additionally, we evaluated our model's effectiveness as a reward model for virtual agents during both the training and inference phases. **1) Training Phase.** Using WebArena and Android World as benchmarks, we employed our model and other reward models to annotate GPT-4o-collected data from these environments, generating preference data. This preference data was then used to perform DPO training on the open-source agents OS-Atlas [37] and UGround [12], validating our model's ability to guide agents in the training phase. **2) Inference Phase.** With Android World and OSWorld as benchmarks, we used OS-Atlas as the open-source agent and GPT-4o-1120 and GPT-4-Turbo as the closed-source agents. During inference, `Similar` and other reward models evaluated the agent's simulated N actions, providing rewards and updating the states of nodes in MCTS. Notably, 30% of the examples partitioned from these benchmarks mentioned earlier were used as the evaluation data.

5.2 Effective Alignment of Preference

We first report the performance of the models on ***SRMEval*** in Table 1. The main findings are as follows: **1)** Effectiveness of step-wise, multi-dimensional, cross-platform data: Using our collected data for reward modeling, `Similar`-RL-Llama achieved an Avg score of 53.9, remarkably outperforming the baseline Llama-3.2-11B-Vision-Instruct with 47.6 ($\uparrow 13.2\%$) and surpassing closed-source models GPT-4o (51.4) and GPT-4-Turbo (46.6). It demonstrates that training reward models with our data enables fine-grained, step-based evaluation, providing a more comprehensive and accurate assessment

Table 2: Task Success Rates (SR) on Android World and WebArena in training setting.

AGENT	REWARD MODEL	ANDROID WORLD SR	WEBARENA SR
GPT-4-TURBO	/	24.1	11.2
GPT-4O	/	25.4	12.7
UGROUND	/	32.4	19.6
UGROUND	QWEN2-VL	32.6	20.2
UGROUND	+ Similar-RL	33.1	26.5
UGROUND	+ Similar-TM	33.9	35.9
UGROUND	LLAMA-3.2-V	33.0	23.4
UGROUND	+ Similar-RL	33.8	29.6
UGROUND	+ Similar-TM	**34.6**	**36.7**
OS-ATLAS	/	30.4	20.2
OS-ATLAS	QWEN2-VL	30.8	20.8
OS-ATLAS	+ Similar-RL	32.1	25.9
OS-ATLAS	+ Similar-TM	34.2	34.5
OS-ATLAS	LLAMA-3.2-V	31.3	22.4
OS-ATLAS	+ Similar-RL	33.6	27.4
OS-ATLAS	+ Similar-TM	**34.9**	**35.6**

Table 3: Task Success Rates (SR) on Android World and OSWorld in inference setting.

AGENT	REWARD MODEL	ANDROID WORLD SR	OSWORLD SR
GPT-4-TURBO	/	24.1	8.4
GPT-4-TURBO	QWEN2-VL	24.9	8.9
GPT-4-TURBO	+ Similar-RL	25.9	10.5
GPT-4-TURBO	+ Similar-TM	28.3	13.4
GPT-4-TURBO	LLAMA-3.2-V	25.3	8.8
GPT-4-TURBO	+ Similar-RL	26.5	10.8
GPT-4-TURBO	+ Similar-TM	**30.4**	**13.9**
GPT-4O	/	25.4	10.8
GPT-4O	QWEN2-VL	26.0	11.3
GPT-4O	+ Similar-RL	27.1	12.9
GPT-4O	+ Similar-TM	32.9	14.3
GPT-4O	LLAMA-3.2-V	26.2	11.7
GPT-4O	+ Similar-RL	29.6	13.1
GPT-4O	+ Similar-TM	**34.6**	**16.5**
OS-ATLAS	/	30.4	14.3
OS-ATLAS	QWEN2-VL	30.9	14.8
OS-ATLAS	+ Similar-RL	32.0	15.4
OS-ATLAS	+ Similar-TM	34.5	16.4
OS-ATLAS	LLAMA-3.2-V	31.5	14.8
OS-ATLAS	+ Similar-RL	32.9	15.7
OS-ATLAS	+ Similar-TM	**35.4**	**17.8**

of GVA action quality. **2)** Synergistic gain from the Triple-M strategy: Similar-TM-Llama achieved an Avg score of 61.2, significantly outperforming Similar-RL-Llama with 53.9 ($\uparrow 13.5\%$). And it achieved higher scores across all dimensions, with improvements such as H increasing from 48.2 to 63.8 ($\uparrow 32.3\%$) and E increasing from 47.1 to 59.2 ($\uparrow 25.6\%$). The Similar-TM-Qwen model showed similar performance. This highlights the effectiveness of our Triple-M strategy, leveraging the complementary expertise of each component to achieve synergistic gain. The experiments demonstrate our model's ability to align preferences.

5.3 Similar for RL Training

We used GPT-4o and multiple reward models to annotate reward data across benchmark environments. The annotated data was then used to train the final agent via DPO. The results, shown in Table 2, demonstrate that our model significantly improves agent learning: **1)** The Similar-RL model derived through Reward Modeling on the ***SRMTrain*** dataset outperforms the baseline. When using OS-Atlas as the agent, Similar-RL-Llama achieves improvements of 10.5% ($30.4 \rightarrow 33.6$) and 7.3% ($31.3 \rightarrow 33.6$) over the original OS-Atlas model and the setting using Llama-3.2V as the reward model, respectively, on Android World. On WebArena, the improvements are 35.6% ($20.2 \rightarrow 27.4$) and 22.3% ($22.4 \rightarrow 27.4$), respectively. **2)** The Similar-TM model performed best. With OS-Atlas, Similar-TM-Llama achieved improvements of 3.8% ($33.6 \rightarrow 34.9$) and 29.9%($27.4 \rightarrow 35.6$) on Android World and WebArena, respectively, compared to Similar-RL-Llama. **3)** When using UGround as the agent or adopting Qwen2-VL as the baseline reward model, comparable performance can be observed. The consistent performance improvements across different models and environments demonstrate that our method enhances virtual agents' learning capabilities.

5.4 Similar for Inference-Time Scaling

During inference, we used various reward models to evaluate the agent's N simulated actions, providing rewards and updating MCTS node states. Table 3 shows that our model effectively guides the agent: **1)** Consistent with the training setup, the Similar-RL model outperformed both the original agent without a reward model and the setting using MLLM as the reward model. With GPT-4o, Similar-RL-Llama achieved improvements of 16.5% ($25.4 \rightarrow 29.6$) and 12.9% ($26.2 \rightarrow 29.6$) on Android World for these two settings, respectively. A similar performance is observed on OSWorld. **2)** The Similar-TM model performed best. With GPT-4o, Similar-TM-Llama achieved improvements of 16.8% ($29.6 \rightarrow 34.6$) and 25.9% ($13.1 \rightarrow 16.5$) on Android World and OSWorld, respectively, compared to Similar-RL-Llama. **3)** When employing GPT-4-Turbo, GPT-4o, or OS-Atlas as the agent, or when using Qwen2-VL as the baseline reward model, we consistently observe similar model performance. It can be concluded that our method is generalizable and effectively enhances the virtual agent's inference ability.

We further demonstrate that our model is essential for scaling the inference-time capabilities of agents by varying the number of child nodes N in MCTS. As shown in Figure 5 (a): 1) When $N \leq 8$, agent performance improves. However, when $N > 8$, performance plateaus or declines, likely due to

Table 4: Ablation study (inference experiments). `Similar` in table represents `Similar`-TM-Llama.

Model	Dimension					Success Rate	
	H	OS	E	TR	C	AW	WA
Backbone						30.4	20.6
+H	✓					32.5	26.1
+OS		✓				31.9	24.7
+E			✓			31.6	23.3
+TR				✓		31.1	21.6
+C					✓	30.9	21.0
+OS,E		✓	✓			32.7	27.5
+H,E	✓		✓			33.1	29.8
+H,OS	✓	✓				33.4	31.4
+TR,C				✓	✓	31.5	22.5
+H,OS,E	✓	✓	✓			34.3	35.9
+OS,E,TR,C		✓	✓	✓	✓	33.1	33.9
+H,E,TR,C	✓		✓	✓	✓	33.9	35.7
+H,OS,TR,C	✓	✓		✓	✓	34.2	36.5
+H,OS,E,C	✓	✓	✓		✓	34.7	37.2
+H,OS,E,TR	✓	✓	✓	✓		35.1	37.7
`Similar`	✓	✓	✓	✓	✓	**35.4**	**38.2**

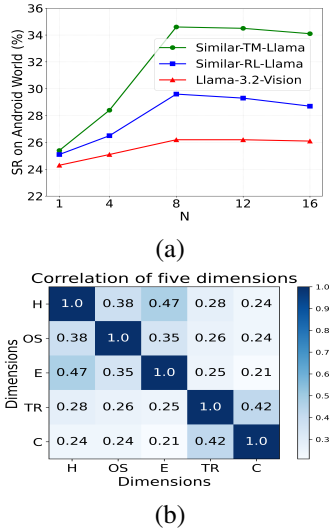


Figure 5: (a) Inference-time scaling research. The agent is GPT-4o. (b) Correlation research of five dimensions.

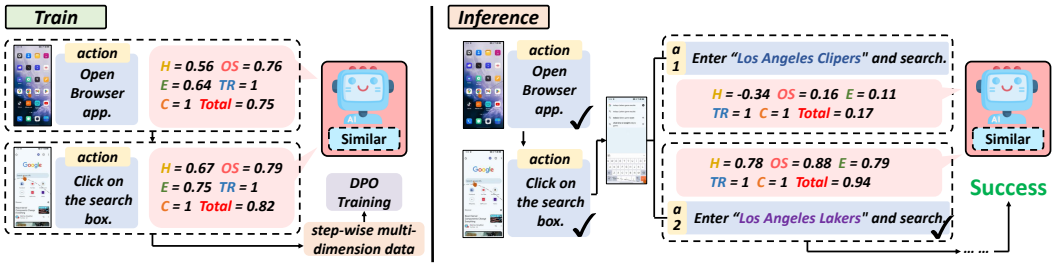


Figure 6: A case of `Similar` provides guidance for GVA training and inference.

limitations in the agent model, as simulating more actions fails to identify viable paths. 2) `Similar`-RL and `Similar`-TM outperform other settings, with **Similar**-RL surpassing MLLM-based reward models and `Similar`-TM exceeding `Similar`-RL. These results demonstrate the superiority of our models while highlighting the challenges of scaling inference-time in agent systems.

5.5 Indepth Analysis

Ablation study. The results, shown in Table 4, show that models with partial-dimensional rewards underperformed compared to `Similar`. For example, on Android World, models excluding H, OS, E, TR, and C rewards showed declines of 6.9%, 4.4%, 3.5%, 2.0%, and 0.8%, respectively, with similar trends on WebArena. Analysis reveals that the H dimension has the most significant impact, as *Helpfulness* captures a step's contribution to task completion. The OS dimension follows closely, reflecting the influence of the current step on the next step. The C dimension has the least impact, as agent actions are often inherently coherent and contextually aligned. These results confirm that fine-grained rewards outperform coarse-grained ones and that our five dimensions comprehensively assess agent actions. More comprehensive results can be found in Appendix E.

Case Study. To demonstrate the role of our model in training and inference, we included visual cases, as shown in Figure 6. During training, `Similar` annotates the agent's trajectory with multi-dimensional scores, used for DPO training. During inference, the agent simulates multiple actions for a single step, and `Similar` evaluates these actions. In the figure, our model assigns high scores to action 2 at the third step, with a total score of 0.94, while action 1 receives lower scores. Therefore, action 2, the highest-scoring action, is easily selected for the current step. More case studies are detailed in Appendix D.

Correlation Study. The Pearson correlation coefficients among the five dimensions are calculated to analyze their independence, as shown in Figure 5 (b). The results show that while some correlation exists among the five dimensions, the values are all below 0.47, indicating independence.

6 Conclusion

In this work, we introduce a novel reward model-based paradigm for training GVAs. Our reward model, `Similar`, provides step-wise, multi-dimensional feedback during GVAs' training and inference, enabling fine-grained assessment. Additionally, we build the first reward model evaluation benchmark called ***SRM***. Extensive experiments demonstrate our model's superior performance on ***SRMEval*** and its effectiveness in guiding GVAs across diverse tasks.

References

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[2] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952.

[3] S.R.K. Branavan, Harr Chen, Luke Zettlemoyer, and Regina Barzilay. Reinforcement learning for mapping instructions to actions. In Keh-Yih Su, Jian Su, Janyce Wiebe, and Haizhou Li, editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 82–90, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[4] S.R.K. Branavan, Luke Zettlemoyer, and Regina Barzilay. Reading between the lines: Learning to map high-level instructions to commands. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre, editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1268–1277, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[5] Wendong Bu, Yang Wu, Qifan Yu, Minghe Gao, Bingchen Miao, Zhenkui Zhang, Kaihang Pan, Yunfei Li, Mengze Li, Wei Ji, Juncheng Li, Siliang Tang, and Yueting Zhuang. What limits virtual agent application? omnibench: A scalable multi-dimensional benchmark for essential virtual agent capabilities, 2025.

[6] Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. A dataset for interactive vision language navigation with unknown command feasibility. In *European Conference on Computer Vision (ECCV)*, 2022.

[7] Filippos Christianos, Georgios Papoudakis, Matthieu Zimmer, Thomas Coste, Zhihao Wu, Jingxuan Chen, Khyati Khandelwal, James Doran, Xidong Feng, Jiacheng Liu, Zheng Xiong, Yicheng Luo, Jianye Hao, Kun Shao, Haitham Bou-Ammar, and Jun Wang. Pangu-agent: A fine-tunable generalist agent with structured reasoning, 2023.

[8] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun

Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[9] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 28091–28114. Curran Associates, Inc., 2023.

[10] Minghe Gao, Wendong Bu, Bingchen Miao, Yang Wu, Yunfei Li, Juncheng Li, Siliang Tang, Qi Wu, Yueting Zhuang, and Meng Wang. Generalist virtual agents: A survey on autonomous agents across digital platforms, 2024.

[11] Minghe Gao, Xuqi Liu, Zhongqi Yue, Yang Wu, Shuang Chen, Juncheng Li, Siliang Tang, Fei Wu, Tat-Seng Chua, and Yueting Zhuang. Benchmarking multimodal cot reward model stepwise by visual program, 2025.

[12] Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*, 2024.

[13] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

[14] Peter C Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy Lillicrap. A data-driven approach for learning to control computers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9466–9482. PMLR, 17–23 Jul 2022.

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

[16] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

[17] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv:2406.18629*, 2024.

[18] Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. In *The Twelfth International Conference on Learning Representations*, 2023.

[19] Juncheng Li, Siliang Tang, Linchao Zhu, Wenqiao Zhang, Yi Yang, Tat-Seng Chua, Fei Wu, and Yueting Zhuang. Variational cross-graph reasoning and adaptive structured semantics learning for compositional temporal grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12601–12617, 2023.

[20] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada, July 2023. Association for Computational Linguistics.

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

[22] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision, 2024.

[23] Kaihang Pan, Wang Lin, Zhongqi Yue, Tenglong Ao, Liyu Jia, Wei Zhao, Juncheng Li, Siliang Tang, and Hanwang Zhang. Generative multimodal pretraining with discrete diffusion timestep tokens. *arXiv preprint arXiv:2504.14666*, 2025.

[24] Kaihang Pan, Siliang Tang, Juncheng Li, Zhaoyu Fan, Wei Chow, Shuicheng Yan, Tat-Seng Chua, Yueting Zhuang, and Hanwang Zhang. Auto-encoding morph-tokens for multimodal llm. *arXiv preprint arXiv:2405.01926*, 2024.

[25] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc., 2023.

[26] Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents, 2024.

[27] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Android in the wild: a large-scale dataset for android device control. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[28] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving AI tasks with chatgpt and its friends in hugging face. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[29] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024.

[30] Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7584–7600, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[31] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022.

[32] Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *The 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

[33] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents, 2023.

[34] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[35] Zihan Wang, Yunxuan Li, Yuexin Wu, Liangchen Luo, Le Hou, Hongkun Yu, and Jingbo Shang. Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7309–7319, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[36] Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi Li, Jian Yang, Wangchunshu Zhou, et al. A comparative study on reasoning patterns of openai's o1 model. *arXiv preprint arXiv:2410.13639*, 2024.

[37] Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024.

[38] Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. Agentgym: Evolving large language model-based agents across diverse environments, 2024.

[39] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024.

[40] Nancy Xu, Sam Masling, Michael Du, Giovanni Campagna, Larry Heck, James Landay, and Monica Lam. Grounding open-domain instructions to automate web support tasks. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1022–1032, Online, June 2021. Association for Computational Linguistics.

[41] An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, et al. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562*, 2023.

[42] Fei Yu, Anningzhe Gao, and Benyou Wang. OVM, outcome-supervised value models for planning in mathematical reasoning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 858–875, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[43] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

[44] Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023.

[45] Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Ziyu Liu, Shengyuan Ding, Shenxi Wu, Yubo Ma, Haodong Duan, Wenwei Zhang, Kai Chen, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2.5-reward: A simple yet effective multi-modal reward model, 2025.

[46] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488, 2022.

[47] Yuanzhao Zhai, Tingkai Yang, Kele Xu, Feng Dawei, Cheng Yang, Bo Ding, and Huaimin Wang. Enhancing decision-making for llm agents via step-level q-value models, 2024.

[48] Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Ufo: A ui-focused agent for windows os interaction, 2024.

[49] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*, 2024.

[50] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *arXiv preprint arXiv:2406.09136*, 2024.

[51] Ruiwen Zhou, Yingxuan Yang, Muning Wen, Ying Wen, Wenhao Wang, Chunling Xi, Guoqiang Xu, Yong Yu, and Weinan Zhang. TRAD: Enhancing llm agents with step-wise thought retrieval and aligned decision. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024.

[52] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *ICLR*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Section **??** (Introduction) of our paper clearly reflect the contributions and scope, including the proposed method, addressed problems, and open-source links.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss the limitations of our work in Section **??** (Conclusion).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our work does not focus on theoretical proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide an anonymous link to the code in the Abstract, enabling result reproduction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: An anonymous link to the code is provided in the Abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Training and test details are provided in Section **??** (Experiments) under "Implementation Details" and "Experimental Setup," with further details in Appendix **??** and Appendix **??**.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experimental results do not include error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resource information for reproducing experiments is provided in Appendix **??**.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research does not violate any NeurIPS ethical guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This is foundational work on modality alignment and does not involve societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not involve high-risk data or models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all referenced works and adhere to their licenses and usage terms.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Detailed documentation for new assets is included in the anonymous link provided in the Abstract.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or human subject research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or human subject research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used

only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: In this paper, we only use large language models (LLMs) to assist with writing, such as polishing sentences and checking grammar.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.