DOES THIS ACTION STILL EXPLAIN THE TASK? REVERSE SCORING FOR DIFFUSION LANGUAGE MODEL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion-based large language models (dLLMs) promise to break the sequential latency bottleneck of autoregressive agents through parallel decoding, but recent evaluations show this efficiency does not transfer to embodied agentic competence: dLLM-backed agents repeatedly fall into retry loops, re-issuing an action long after it has failed. We give a mechanistic account of this failure and a training-free remedy. We trace the retry loop to the adaptivity of masked decoding: the sampler commits the positions it is most confident about and defers the uncertain ones, and at a failure state the context already offers a confident fill for the deferred decision, i.e. the failed action itself, so the retry is committed without the failure feedback ever being confronted. We model the resulting distortion of the action distribution as a *task-blind corruption*: contextually salient actions (e.g., the action just taken) receive inflated probability by a factor that depends on the state and the action but not on the task. Under this model, we analyse an invariance proposition: the task-blind factor cancels exactly from the *reverse conditional*, i.e. the likelihood of the task given the state and a candidate action, which coincides with the task posterior of an idealized uncorrupted model. Masked dLLMs evaluate the reverse conditional natively, unlike autoregressive models, by masking the task tokens and denoising, at the cost of a few parallel passes per candidate. We instantiate the rule as **Reflect Reverse** and evaluate it on four multi-turn embodied benchmarks, where it improves task success and progression rates over forward-scoring baselines.

1 INTRODUCTION

An agent that samples its next action from a language model inherits the pathologies of the distribution it samples from. Diffusion language models (dLLMs) are attractive agent backbones: they generate entire spans in a handful of parallel denoising steps, promising to break the sequential latency bottleneck of autoregressive decoding. But the distribution a dLLM samples from has a distinctive, structural pathology in embodied settings: it over-weights actions that are already salient in the context, by an amount that depends on the state and the action but not on the task. The bias is present at every decision point, however it is clearly visible on failure states: at a state where the previous action has just failed, the model keeps proposing that same action, verbatim, as if the most re-usable prediction were whatever already sits in context.

Why should this be structural rather than a matter of knowledge? Masked decoding is *adaptive*: the sampler commits the masked positions it is most confident about and defers the rest (Nie et al., 2025), and Ni et al. (2026) show that the deferred positions are the "forking" ones where several continuations remain viable: by the time the sampler returns to them, the committed context has already resolved the choice. At a failure state, *which action now?* is such a fork, and the context already offers a confident fill for it: the failed action itself, verbatim in s. The retry is thus committed without the open decision ever being sampled: the failure feedback is bridged, not confronted (Section 5 develops the mechanism). Resampling at a stuck state shows the knowledge is there: the candidates include plausible alternatives, yet the distribution still concentrates on the contextually salient one. The bias does not care which task is being attempted.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

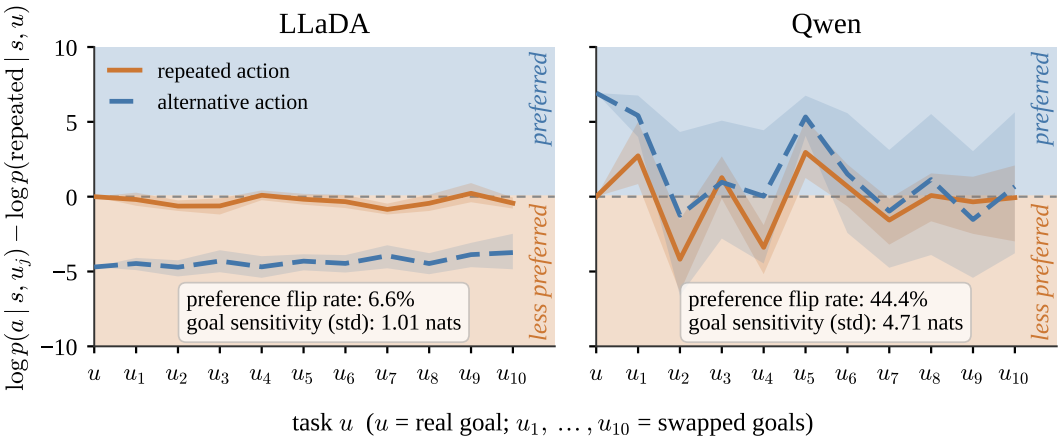

Figure 1: Task-blind corruption analyzed over 208 failed states in ALFWorld.

We make this precise and turn it into a correction with a provable guarantee. Let u denote the task, s the interaction history, and a a candidate next action. Write $\pi_\theta(a \mid s, u)$ for the observable model conditional, and $\mu(a \mid s, u)$ for the *faithful reference* conditional: the distribution the model would produce if it had absorbed s correctly. Our hypothesis is that the gap between the two is a multiplicative distortion that does not depend on the task (Assumption 2.1): the same state–action inflation must explain the model's behavior under every goal. To verify it empirically, we record the log-probability of the retried ("repeated") and goal-directed ("alternative") actions under the real goal u and ten swapped goals $u_1, \ldots, u_{10}$, and plot the mean change relative to the real-task level (Figure 1). Full details of the task-corruption procedure are provided in Appendix A. For both actions, LLaDA (Nie et al., 2025) is nearly insensitive to goal swaps, and always prefers the failed repeated action, in contrary to the faithful reference represented by Qwen here.

If the corruption is task-blind, it should be invisible to any score that compares candidates *by their task evidence alone*. This suggests inverting the question. The standard rule asks the model the forward question: *which action for this task?* — and the corruption survives, because it inflates an action's probability independently of the task. We instead ask the reverse question: *does the state, together with this candidate action, still explain the task?* Every candidate is then scored on the task evidence for it, $\pi_\theta(u \mid s, a)$; under the coherence idealization of Section 2.2, a factor that carries no task information cannot move this score. For a masked dLLM the reverse question is a native operation unavailable easily in autoregressive models: mask the task tokens, make the state and the candidate action visible and denoise.

Our main theoretical analysis makes this precise (Proposition 3.1): under the corruption and coherence assumptions (Section 2.2), the reverse score $\pi_\theta(u \mid s, a)$ equals, for every action, the faithful model's task posterior under a task prior that the corruption reweights. The posterior is proportional to a *task-evidence ratio* $\mu(a \mid s, u)/m(a)$: the faithful likelihood of the action under the actual task, divided by its likelihood $m(a)$ under a mixture of tasks. The corruption cannot survive this comparison: it varies across actions, which is how it distorts the forward ranking, but at each action it is constant across tasks, which is why it cancels from a posterior over tasks.

We instantiate our method as **Reflect Reverse**: sample candidates with chain of thoughts, score each by the reverse conditional, sample the next action proportionally to the scores. On four multi-turn embodied benchmarks and two dLLMs, Reflect Reverse reduces retry-loops and improves task success and progression rates over forward-scoring baselines.

Contributions.

- A task-blind corruption hypothesis for dLLM action distributions, stated formally and empirically verified (Section 1, Figure 1).
- A recovery proposition: the reverse score equals the faithful model's task posterior (Proposition 3.1); we also analyze the reranking's residual bias (Theorem 3.1).

- Reflect Reverse, a training-free action-selection rule that scores candidates by the model's reverse conditional, requiring only a few forward passes per candidate (Section 3).
- Experiments on ALFWorld, ScienceWorld, BabyAI, and Jericho showing improved success rates and progression rates with LLaDA and iLLaDA (Section 4).

2 BACKGROUND

2.1 MASKED DIFFUSION LANGUAGE MODELS

Masked dLLMs (Nie et al., 2025) define a forward process that progressively masks tokens of a sequence $x = (x_1, \ldots, x_n)$ and a reverse process that restores them. At each reverse step, a Transformer with bidirectional attention observes the currently unmasked tokens x^o and predicts every masked position simultaneously $p_\theta(x^m \mid x^o) = \prod_{j \in m} p_\theta(x_j \mid x^o)$, i.e., the per-step joint over masked tokens is *factorized* given the context. Any span of a sequence can be scored against any other span by masking the first and clamping the second, in multiple denoising steps; there is no privileged direction of conditioning.

2.2 TASK-BLIND CORRUPTION

We model the observed conditional as a faithful conditional distorted by a state–action factor.

Assumption 2.1 (Task-blind multiplicative corruption)**.** *There exists a function $\varepsilon(s, a)$, depending on the state and the action but* not *on the task, such that for every task u in the support of $p(\cdot \mid s)$:*

$$\pi_\theta(a \mid s, u) = \frac{\mu(a \mid s, u)\, e^{\varepsilon(s,a)}}{Z(s, u)}, \qquad Z(s, u) := \sum_{a'} \mu(a' \mid s, u)\, e^{\varepsilon(s,a')}. \tag{1}$$

The content of the assumption is the clause that ε does not depend on the task: for any two conditionals one can always fit some ε. Figure 1 tests it: at fixed states, the log-ratio $\log \pi_\theta(a \mid s, u) - \log \mu$-proxy varies with (s, a) but is flat in u. The factor $\varepsilon(s, a)$ is the structural bias itself: an action whose tokens are salient in s is inflated by the same amount whatever the goal.

3 METHOD

3.1 REFLECT REVERSE

Standard decoding samples an action from $\pi_\theta(a \mid s, u)$, i.e. the forward question, but inherits the corruption of Assumption 2.1. Reflect Reverse inverts the direction of conditioning:

1. **Propose.** Sample K candidate actions $a_1, \ldots, a_K \sim \pi_\theta(\cdot \mid s, u)$ with chain of thoughts and deduplicate, yielding a candidate set $\mathcal{A}(s)$.
2. **Reflect.** Score each candidate by the reverse conditional, estimated by Monte Carlo rollouts: provide s and a, sample M random masks of the task tokens, denoise, and read off the task-token log-likelihoods at the masked positions:

$$\phi(a) := \frac{1}{M} \sum_{m=1}^{M} \frac{1}{|B_m|} \sum_{j \in B_m} \log \pi_\theta(u_j \mid s, a, \tilde{u}^{(m)}), \tag{2}$$

   where $B_m \subseteq \{1, \ldots, |u|\}$ is the random subset of task-token positions masked in rollout m and $\tilde{u}^{(m)}$ is the masked task-token. Each term of Eq.2 is a masked-estimate of the task log-likelihood $\log \pi_\theta(u \mid s, a)$, so $\phi(a)$ is a Monte Carlo estimate of the same reverse score.
3. **Act.** Sample the next action from the corrected policy

$$\rho(a \mid s, u) := \frac{\exp(\beta\, \phi(a))}{\sum_{a' \in \mathcal{A}(s)} \exp(\beta\, \phi(a'))}, \qquad a \in \mathcal{A}(s), \tag{3}$$

   with inverse temperature $\beta > 0$.

The whole procedure is training-free and adds $|\mathcal{A}(s)| \times M$ denoising passes per decision. Pseudocode is in Appendix B.

3.2 INVARIANCE OF THE REVERSE SCORE TO TASK-BLIND CORRUPTION

The reverse score is not an arbitrary heuristic: under the corruption model of Section 2.2, it equals the faithful model μ's own verdict.

Proposition 3.1 (Recovery of the faithful task posterior)**.** *Let Assumptions 2.1 and C.1 hold. Then, for every state s, task u in the support of $p(\cdot \mid s)$, and action a,*

$$\pi_\theta(u \mid s, a) \;=\; \frac{\omega(u \mid s)\,\mu(a \mid s, u)}{\sum_{u'} \omega(u' \mid s)\,\mu(a \mid s, u')}, \tag{4}$$

where $\omega(u \mid s) := p(u \mid s)/Z(s, u)$ is a positive task weight that does not depend on the action a.

The proof is in Appendix C.1. The cancellation is the pointwise mutual information (PMI) invariance identified by Holtzman et al. (2021). What is new here is not the identity but its agentic instantiation. The corruption is not an input-level bias corrected by an external signal (Holtzman et al., 2021; Chae et al., 2024) but a decoding-level mechanism.

An exact score for comparing candidates. Identity Eq.4 holds pointwise in a, so the pairwise log-margins of the score $\pi_\theta(u \mid s, \cdot)$ coincide with those of the faithful task-evidence ratio $\mu(a \mid s, u)\big/\sum_{u'} \omega(u' \mid s)\,\mu(a \mid s, u')$: the score orders and ties candidates exactly as the ratio does. The remaining factor $\omega(u \mid s)$ is shared by all candidates, so softmax normalization removes it at any temperature (Corollary C.1): **the corrected policy ρ is exactly the softmax of the faithful task-evidence ratio over the candidate set, at any β.**

What the guarantee does and does not cover. The guarantee governs the comparison among the candidates on the table, not the table itself: the proposal step still draws candidates from the corrupted $\pi_\theta(\cdot \mid s, u)$, so the support $\mathcal{A}(s)$ need not coincide with the faithful model's support. Moreover, ρ matches the faithful *task-evidence ratio*, not the normalized faithful conditional: writing $m(a) := \sum_{u'} \omega(u' \mid s)\,\mu(a \mid s, u')$ for the task-marginal likelihood of a, we have $\rho(a) \propto \mu(a \mid s, u)^\beta \big/ m(a)^\beta$, and the factor $m(a)$ depends on the candidate and is not removed by normalization. ρ therefore coincides with sampling from $\mu(\cdot \mid s, u)$ restricted to $\mathcal{A}(s)$ only when the marginal is flat across the candidate set (Remark C.1); what holds unconditionally is the exact recovery of the ratio's ordering and ties.

Why the forward score cannot say this. Raw probability has no such invariance: $\pi_\theta(a \mid s, u) \propto \mu(a \mid s, u)\, e^{\varepsilon(s,a)}$ reorders freely as ε varies across candidates. The reverse question is precisely the one in which a task-blind factor is invisible under our assumptions. At a failure state, the retried action carries a large $\varepsilon(s, a)$, because it is already in context, so it wins the forward comparison; but its faithful task evidence is low, because it just failed, so it loses the reverse comparison.

3.3 HOW LARGE IS THE RESIDUAL BIAS?

How far is ρ from the faithful reference $\mu(\cdot \mid s, u)$ on the candidate set, and how does that residual bias compare with the bias of forward selection from the corrupted $\pi_\theta(\cdot \mid s, u)$ on the same set? Theorem 3.1 below answers both questions in total variation, $d_{\mathrm{TV}}(P, Q) := \frac{1}{2}\sum_a |P(a) - Q(a)|$; the proof, and the sharp range lemma it rests on, are in Appendix C.2.

Fix a state s, a task u in the support of $p(\cdot \mid s)$, and a finite candidate set $\mathcal{A} \subseteq \operatorname{supp} \mu(\cdot \mid s, u)$, abbreviating $\mathcal{A} = \mathcal{A}(s)$ when the candidate set is the proposed one. Write $\mu_{\mathcal{A}}$ and $\pi_{\mathcal{A}}$ for the faithful and corrupted conditionals renormalized to $\mathcal{A}$ and measure the heterogeneity of the candidates through the log-spreads:

$$\Lambda_\mu \;:=\; \max_{a \in \mathcal{A}} \log \mu(a \mid s, u) - \min_{a \in \mathcal{A}} \log \mu(a \mid s, u), \tag{5}$$

$$\Lambda_m \;:=\; \max_{a \in \mathcal{A}} \log m(a) - \min_{a \in \mathcal{A}} \log m(a), \tag{6}$$

$$\Lambda_{\mathrm{task}} \;:=\; \sup_{u' \in \operatorname{supp} p(\cdot \mid s)} \Big(\max_{a \in \mathcal{A}} \log \mu(a \mid s, u') - \min_{a \in \mathcal{A}} \log \mu(a \mid s, u') \Big) \;\in\; [0, +\infty], \tag{7}$$

where the supremum runs over tasks u' under which $\mu(\cdot \mid s, u')$ is not identically zero on $\mathcal{A}$, with the convention $\log 0 = -\infty$. The spreads Λ_μ and Λ_m are always finite, and Λ_{task} is finite in particular whenever no candidate is ruled out by any task in the support.

Theorem 3.1 (Reverse selection is bounded by the faithful spread; forward selection by the corruption)**.** *Fix a state s, a task u in the support of $p(\cdot \mid s)$, and a finite candidate set $\mathcal{A} \subseteq \operatorname{supp} \mu(\cdot \mid s, u)$ with $|\mathcal{A}| \geq 2$. Let Assumptions 2.1 and C.1 hold and let $\beta > 0$. Let $a^+ := \arg\max_{a \in \mathcal{A}} \varepsilon(s, a)$ be the most inflated candidate, $\gamma := \mu_{\mathcal{A}}(a^+)$ its faithful mass, and*

$$\Delta_2 \;:=\; \varepsilon(s, a^+) - \max_{a \in \mathcal{A} \setminus \{a^+\}} \varepsilon(s, a) \;\geq\; 0 \tag{8}$$

the inflation gap between the most inflated candidate and the rest. Then:

(i) ***The reverse bias is bounded independently of the corruption.***

$$d_{\text{TV}}(\rho, \mu_{\mathcal{A}}) \;\leq\; \tanh\Big(\frac{|\beta - 1|\,\Lambda_\mu + \beta \Lambda_m}{4}\Big) \;\leq\; \tanh\Big(\frac{|\beta - 1|\,\Lambda_\mu + \beta \Lambda_{\text{task}}}{4}\Big), \tag{9}$$

and the right-most bound does not involve ε at all; at $\beta = 1$ it reads $d_{\text{TV}}(\rho, \mu_{\mathcal{A}}) \leq \tanh(\Lambda_m/4) \leq \tanh(\Lambda_{\text{task}}/4)$.

(ii) ***The forward bias grows with the corruption.***

$$d_{\text{TV}}(\mu_{\mathcal{A}}, \pi_{\mathcal{A}}) \;\geq\; \frac{\gamma(1-\gamma)\big(1 - e^{-\Delta_2}\big)}{\gamma + (1-\gamma)e^{-\Delta_2}}, \tag{10}$$

with equality when $|\mathcal{A}| = 2$; the right-hand side is increasing in Δ_2 and tends to $1 - \gamma$ as $\Delta_2 \to \infty$.

Part (i) controls the residual bias of the corrected policy by the spread of the *faithful* log-probabilities over the candidate set: at $\beta = 1$ the bound is $\tanh(\Lambda_{\text{task}}/4)$, a quantity of the uncorrupted model, whatever the size of ε. Part (ii) shows that the forward-selection bias is a function of the corruption itself: it grows with the inflation gap Δ_2 between the most inflated candidate and the rest, toward $1 - \gamma$. Where the corruption is flat, forward selection is exact and the corrected policy retains only the m-bias; the retry loop is the regime in which the two are farthest apart, with forward selection's bias growing while the corrected policy's stays capped by a corruption-free quantity.

4 Experiments

4.1 Experimental setup

Benchmarks. We evaluate our proposed method on four multi-turn agent benchmarks spanning diverse task domains: ALFWorld (Shridhar et al., 2020) for household task execution, ScienceWorld (Wang et al., 2022) for interactive scientific reasoning, BabyAI (Chevalier-Boisvert et al., 2018) for instruction following in gridworld environments, and Jericho (Hausknecht et al., 2019) for text-based adventure games.

Backbone models and inference. We use LLaDA-8B-Instruct (Nie et al., 2025) and iLLaDA-8B-Instruct (Nie et al., 2026) as the backbone models for our dLLM-based agents. iLLaDA follows LLaDA's masked diffusion formulation and is trained from scratch with larger-scale pre-training and supervised fine-tuning. For efficient inference, we use the Fast-dLLM (Wu et al., 2025) implementation for LLaDA and adapt its block-wise approximate key–value (KV) caching and parallel decoding to iLLaDA.

Baseline. We compare our proposed method, **Reflect Reverse**, with **One Pass**. Following the action selection procedure of Lu et al. (2026), this baseline executes the first valid action extracted from the generated output.

Evaluation protocol. We evaluate both methods at two sampling temperatures $T \in \{0.8, 0.9\}$ for LLaDA and $T \in \{0.9, 1.0\}$ for iLLaDA, with three random seeds 12, 22 and 32. Each evaluation episode is limited to 30 interaction steps and terminates early if the task is completed. At each step,

we generate 20 candidate actions and apply the corresponding action-selection method. Generated actions are mapped to valid environment actions through similarity-based matching, with a match accepted only when its similarity score exceeds 0.5. Complete hyperparameters for the experiments are summarized in Appendix D.

Evaluation metrics. For each benchmark, we report success rate (SR) and progress rate (PR) to capture complementary aspects of agent performance. Success rate measures the fraction of evaluation episodes in which the agent completes the task within the interaction budget. Progress rate measures advancement toward the task objective, averaged across evaluation episodes.

4.2 Main results

Figure 2 compares **Reflect Reverse** with **One Pass** across the four benchmarks and two backbone models. **Reflect Reverse** achieves higher mean SR in 13 of the 16 evaluation groups and higher mean PR in 14. In particular, SR improves on ALFWorld, ScienceWorld, and BabyAI for both backbones at both evaluated temperatures, while PR improves in all eight configurations for iLLaDA.

Notably, SR improves on ALFWorld, ScienceWorld, and BabyAI for both backbones at both evaluated temperatures. For iLLaDA, PR also improves across all four benchmarks at both temperatures, demonstrating consistent gains in advancement toward task objectives.

The improvements are substantial in several settings. For LLaDA, the largest absolute SR gain occurs on ALFWorld at $T = 0.9$, where SR increases from 8.71% to 16.17%, accompanied by an increase in PR from 29.33% to 36.24%. For iLLaDA, the largest absolute SR gain occurs on BabyAI at $T = 1.0$, where SR rises from 11.61% to 20.24% and PR increases from 25.13% to 35.40%.

The benefits also extend to partial progress on Jericho, where PR improves for both backbones at both temperatures, although SR decreases at the lower temperature evaluated for each backbone. Across all configurations, the only PR decreases occur in two LLaDA settings and are smaller than one percentage point in both cases. Taken together, these results support reverse reflection as an effective approach to improving multi-turn agent performance, with gains spanning both backbone models and diverse task domains.

4.3 Ablations

4.3.1 Rerank on forward score

To evaluate the contribution of reverse conditional scoring, we compare **Reflect Reverse** with **Reflect Forward**. Rather than masking task description tokens u as in Eq.2, **Reflect Forward** masks candidate action tokens a and scores their reconstruction conditioned on s, u, and the unmasked action tokens:

$$\psi(a) := \frac{1}{M} \sum_{m=1}^{M} \frac{1}{|B_m|} \sum_{j \in B_m} \log \pi_\theta(a_j \mid s, u, \tilde{a}^{(m)}), \tag{11}$$

where $B_m \subseteq \{1, \ldots, |a|\}$ is a randomly sampled, nonempty set of masked positions, and $\tilde{a}^{(m)}$ denotes the candidate action with the tokens at those positions masked. The Act step retains the same softmax action-selection policy, replacing $\phi(a)$ with $\psi(a)$ when sampling the next action.

Table 1 compares the mean success rate and progress rate across four benchmarks, averaged over two sampling temperatures, and reports the relative change (Δ) between **Reflect Reverse** and **Reflect Forward**. Consistent with these averaged results, the detailed results in Tables 3 and 4 in Appendix E show that **Reflect Reverse** generally outperforms **Reflect Forward**. Across the 16 configurations, reverse scoring achieves higher SR in 12 and matches it at the reported precision in three others. It also achieves higher PR in 14 configurations.

For LLaDA, reverse scoring improves both SR and PR on ALFWorld and ScienceWorld at both $T = 0.8$ and $T = 0.9$. On ALFWorld at $T = 0.9$, SR increases from 8.96% to 16.17% and PR rises from 31.45% to 36.24%. On Jericho, reverse scoring matches forward scoring in SR at both temperatures while improving PR from 17.52% to 20.53% at $T = 0.8$ and from 20.94% to 26.72% at $T = 0.9$.

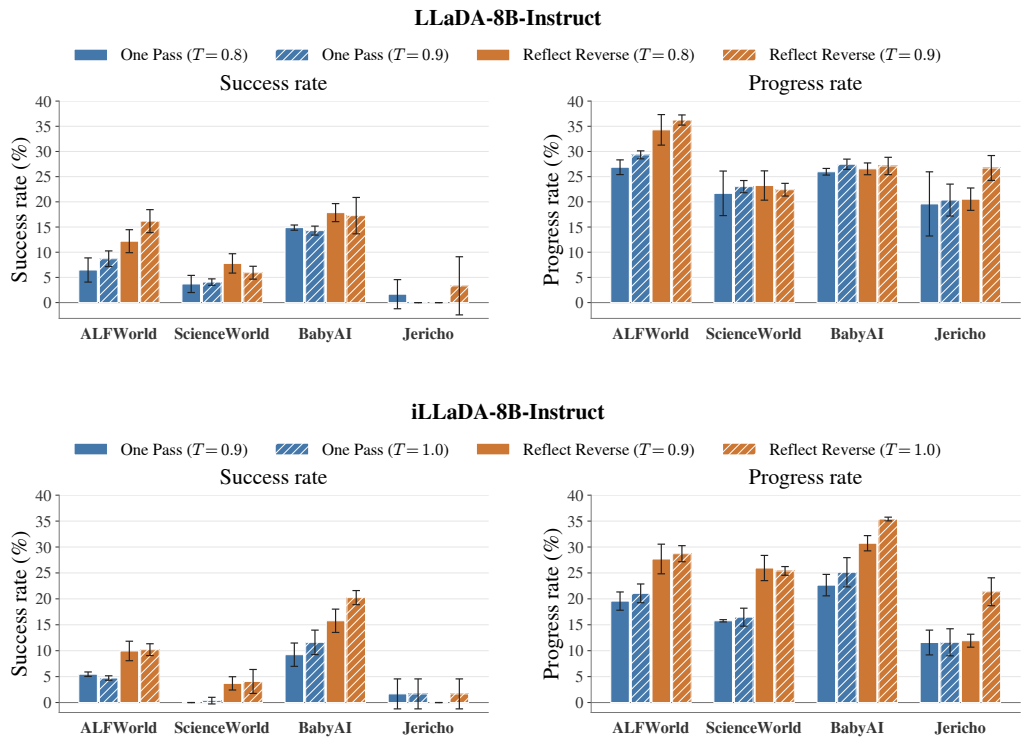

Figure 2: Success rate and progress rate for LLaDA-8B-Instruct and iLLaDA-8B-Instruct.

Table 1: Comparison of success rate (SR) and progress rate (PR) for **Reflect Reverse** and **Reflect Forward**, averaged across the two sampling temperatures. Reverse and Forward results are reported as mean ± standard deviation (%). The relative change is computed as $\Delta = (\text{Reverse} - \text{Forward})/\text{Forward} \times 100\%$. Positive relative changes are shown in bold. When the Forward value is zero, the relative change is undefined and reported as N/A.

Model	Benchmark	Success Rate (SR)			Progress Rate (PR)		
		Reverse	Forward	Δ (%)	Reverse	Forward	Δ (%)
LLaDA	ALFWorld	14.18 ± 2.99	8.21 ± 1.33	**+72.72**	35.26 ± 2.29	30.59 ± 1.79	**+15.27**
	ScienceWorld	6.85 ± 1.78	4.63 ± 1.78	**+47.95**	22.82 ± 2.06	21.15 ± 2.43	**+7.90**
	BabyAI	17.56 ± 2.57	17.56 ± 3.32	0.00	26.83 ± 1.35	29.40 ± 2.35	−8.74
	Jericho	1.67 ± 4.08	1.67 ± 2.58	0.00	23.63 ± 3.99	19.23 ± 2.61	**+22.88**
iLLaDA	ALFWorld	10.07 ± 1.40	0.75 ± 0.94	**+1242.67**	28.20 ± 2.13	16.08 ± 1.56	**+75.37**
	ScienceWorld	3.89 ± 1.69	0.19 ± 0.45	**+1947.37**	25.67 ± 1.66	8.76 ± 1.68	**+193.04**
	BabyAI	18.01 ± 2.96	8.48 ± 1.23	**+112.38**	33.07 ± 2.73	19.98 ± 2.42	**+65.52**
	Jericho	0.83 ± 2.04	0.00 ± 0.00	N/A	16.65 ± 5.50	10.57 ± 1.49	**+57.52**

The gains are particularly consistent for iLLaDA, where reverse scoring improves SR in seven of eight configurations and PR in all eight. On BabyAI, SR increases from 8.93% to 15.77% at $T = 0.9$ and from 8.04% to 20.24% at $T = 1.0$, while PR rises from 21.52% to 30.73% and from 18.44% to 35.40%, respectively. For LLaDA on BabyAI, reverse scoring improves SR at $T = 0.9$ but is less consistent on PR. Overall, these results support reverse conditional scoring as an effective candidate-selection criterion, with advantages over forward scoring across both backbones and the sampling temperatures evaluated for each.

4.3.2 RETRY STATE LOOPS

The dominant retry-loop patterns consist of repeatedly executing the same action (e.g., `AAAAAA`) or alternating between two actions (e.g., `ABABABAB`). Examples of both pattern can be found in Appendix F. We count each additional complete occurrence beyond the initial action or action pair as one retry, without double-counting overlaps. Thus, `AAAAAA` contains five retries, while `ABABABAB` contains three.

The retry percentage is defined as the total number of retries divided by the interaction budget. As shown in Figure 3, repetitive behavior is strongly associated with sampling temperature: lower temperatures yield higher mean retry percentages in 21 of the 24 pairwise temperature comparisons, suggesting that reduced action diversity increases agent's repetitive behavior. Despite this general trend, **Reflect Reverse** consistently mitigates repetition, achieving the lowest mean retry percentage in all 16 evaluated settings. By contrast, **Reflect Forward** increases repetition relative to **One Pass** in five of the eight iLLaDA settings.

While these results show that **Reflect Reverse** is effective at mitigating repetitive behavior, lower retry frequency does not necessarily translate directly into better task performance. For example, on ALFWorld at $T = 0.9$, iLLaDA with **Reflect Forward** exhibits fewer retries than **One Pass**, yet achieves lower success and progress rates. This indicates that avoiding repetitive action loops is only one component of effective agent behavior: successful task execution also depends on the model's ability to generate correct and task-appropriate actions.

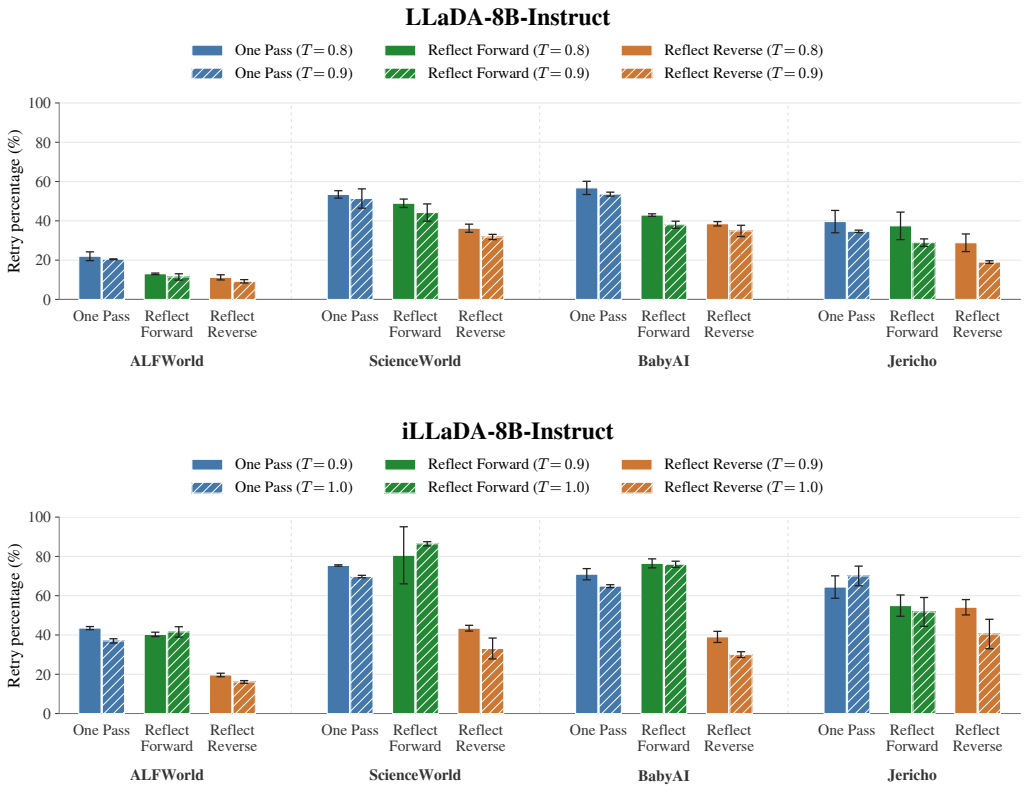

Figure 3: Retry state percentage of LLaDA-8B-Instruct and iLLaDA-8B-Instruct.

5 RELATED WORK

Diffusion language models and adaptive decoding. Masked dLLMs such as LLaDA (Nie et al., 2025) and iLLaDA (Nie et al., 2026) generate by iteratively denoising masked spans; the standard inference recipe couples confidence-based remasking with semi-autoregressive block decoding (Nie et al., 2025; Wu et al., 2025). Ni et al. (2026) identify a failure mode of this recipe in reasoning

tasks: because the sampler commits the positions it is most confident about and defers the rest, it systematically bypasses the high-entropy "forking" tokens at which reasoning paths diverge; by the time these positions are filled, the committed context has already resolved them, so an open decision degenerates into a retrospective alignment with a pre-determined gap (*entropy degradation*). Forcing left-to-right order restores the confrontation of forks and enlarges the reachable solution space, with the effect growing monotonically in the degree of order freedom. The retry loops we study are this trap in a sharper form. At a failure state, *which action now?* is a fork, and unlike in reasoning tasks the context already contains a confident fill for it: the failed action, verbatim in s. Adaptive decoding commits the copy without the open decision ever being sampled, and the deferred positions are finally filled in alignment with the retry.

LLM agents in embodied environments. Reasoning-and-acting agent frameworks (Yao et al., 2023; Shinn et al., 2023; Mower et al., 2026) couple chain-of-thought traces with environment actions and are standardly evaluated on ALFWorld (Shridhar et al., 2020), ScienceWorld (Wang et al., 2022), BabyAI (Chevalier-Boisvert et al., 2018), and Jericho (Hausknecht et al., 2019). Recent evaluations of dLLM-backed agents (Lu et al., 2026) report that the latency gains of parallel decoding do not transfer to agentic competence, with retry loops as a dominant failure mode. We give this failure a mechanistic account, i.e. adaptive decoding bridging the failure feedback instead of confronting it, and a training-free correction.

Sample-then-rerank action selection. Reflect Reverse belongs to the family of decoding-time corrections: best-of-N selection with verifiers (Ji et al., 2025; Kang et al., 2026), self-consistency (Wang et al., 2023), and PMI-style rescoring of raw likelihoods (Holtzman et al., 2021). These methods share our observation that raw model probability is an unreliable score, but they correct it with an external signal: a learned verifier, a majority vote, or a domain-conditional baseline. Our score instead is the model's own reverse conditional $\pi_\theta(u \mid s, a)$. The same reverse-conditional identity appears, in a different role, in Lian et al. (2026), who train vision–language–action policies by maximizing the log-likelihood ratio $\log p(\ell \mid a, v) - \log p(\ell \mid v)$, the conditional PMI between action and instruction, as a training objective; our correction instead is a training-free, decoding-time rule.

6 Conclusion And Future Works

We traced the retry loops that stall diffusion language model agents to a structural property of masked decoding: because the sampler commits the positions it is most confident about and defers the rest, the open decision at a failure state is postponed until the committed context resolves it; and the context already offers a confident fill, the failed action itself. We formalized the resulting distortion of the action distribution as a *task-blind corruption*, i.e. a state–action factor that inflates contextually salient actions independently of the task, and verified it through goal-swap interventions. The formalization is what motivates the correction: a factor that carries no task information cannot survive a posterior over tasks, so it cancels from the reverse conditional $\pi_\theta(u \mid s, a)$, which masked dLLMs evaluate natively by masking the task tokens and denoising. The resulting rule, **Reflect Reverse**, is training-free, requires no learned critic, value function, or environment rollouts, adds only a few parallel denoising passes per candidate, and improves success and progress rates over forward-scoring baselines on four embodied benchmarks and two backbones.

Future work. Firstly, Proposition 3.1 governs the comparison among the candidates on the table but proposals are still drawn from the corrupted forward conditional, so the faithful model's support may go uncovered. Extending the invariance to the proposal step (e.g. dividing out an estimated $\varepsilon(s, a)$) would close this gap. Secondly, the corruption is ultimately a training artifact of adaptive decoding; decoding-time correction could be complemented by fine-tuning at failure states or by regularizing forward–reverse consistency, removing the pathology at its source. Finally, the retry loop is the agentic form of entropy degradation, and the reverse question, *does this continuation still explain the prompt?*, applies to reasoning chains, suggesting reverse scoring as a general decoding-time instrument for masked diffusion models.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, unifying figure styles, proofreading, implementing model dual cache and checking code correctness. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on improving the reliability of diffusion language model agents by reducing repetitive action failures in multi-turn settings. Our experiments are conducted in controlled benchmark environments and do not involve human participants or sensitive personal data. We do not identify any direct harmful applications arising from the proposed method, and any broader risks are those generally associated with the use of language-model agents.

REPRODUCIBILITY STATEMENT

We will publicly release the complete code upon acceptance. All benchmarks used in our experiments, including ALFWorld, ScienceWorld, BabyAI, and Jericho, are publicly available. All evaluated language models, including LLaDA-8B-Instruct, iLLaDA-8B-Instruct, and Qwen2.5-7B-Instruct, are also open source. Section 4.1 describes the benchmarks, backbone models, baselines, evaluation metrics, and experimental protocol. The complete set of hyperparameters required to reproduce our experiments is provided in Appendix D. For the theoretical component, we explicitly specify all notation and assumptions, including the task-blind multiplicative corruption and coherent conditionals assumptions, in Section 2.2. Complete derivations and intermediate steps are provided in Appendix C to facilitate independent verification of the analysis.

REFERENCES

Olivier Binette. A note on reverse pinsker inequalities. *IEEE Transactions on Information Theory*, 65(7):4094–4096, 2019. ISSN 1557-9654. doi: 10.1109/tit.2019.2896192. URL http://dx.doi.org/10.1109/TIT.2019.2896192.

Kyubyung Chae, Jaepill Choi, Yohan Jo, and Taesup Kim. Mitigating hallucination in abstractive summarization with domain-conditional mutual information. In *Findings of the association for computational linguistics: NAACL 2024*, pp. 1809–1820, 2024.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations*, 2018. URL https://api.semanticscholar.org/CorpusID:59536625.

Matthew J. Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. Interactive fiction games: A colossal adventure. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:202565447.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7038–7051, 2021.

Xiaotong Ji, Shyam Sundhar Ramesh, Matthieu Zimmer, Ilija Bogunovic, Jun Wang, and Haitham Bou Ammar. On almost surely safe alignment of large language models at inference-time, 2025. URL https://arxiv.org/abs/2502.01208.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *Advances in neural information processing systems*, 38:19720–19745, 2026.

Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T. Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. Langforce: Bayesian decomposition of vision language action models via latent action queries, 2026. URL https://arxiv.org/abs/2601.15197.

Qingyu Lu, Liang Ding, Kanjian Zhang, Jinxia Zhang, and Dacheng Tao. The bitter lesson of diffusion language models for agentic workflows: A comprehensive reality check. *ArXiv*, abs/2601.12979, 2026. URL https://api.semanticscholar.org/CorpusID:284910491.

Christopher E. Mower, Yuhui Wan, Hongzhan Yu, Antoine Grosnit, Jonas Gonzalez-Billandon, Matthieu Zimmer, Puze Liu, Daniel Palenicek, Davide Tateo, Jan Peters, Kaixian Qu, Mike Zhang, Guowei Lan, Andrei Cramariuc, Cesar Cadena, Marco Hutter, Guangjian Tian, Yuzhen Zhuang, Kun Shao, Xingyue Quan, Jianye Hao, Jun Wang, and Haitham Bou-Ammar. A robot operating system framework for using large language models in embodied AI. *Nature Machine Intelligence*, 8(3):313–325, 2026. ISSN 2522-5839. doi: 10.1038/s42256-026-01186-z. URL https://doi.org/10.1038/s42256-026-01186-z.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, and Gao Huang. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models, 2026. URL https://arxiv.org/abs/2601.15165.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Jirong Wen, and Chongxuan Li. Large language diffusion models. *ArXiv*, abs/2502.09992, 2025. URL https://api.semanticscholar.org/CorpusID:276395038.

Shen Nie, Qi-Yang Min, Shaoxuan Xu, Zihao Huang, Yuxuan Song, Shan Yong, Yankai Lin, Wayne Xin Zhao, Chongxuan Li, and Jiaxin Wen. Improved large language diffusion models. *ArXiv*, abs/2606.25331, 2026. URL https://api.semanticscholar.org/CorpusID:289630838.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *ArXiv*, abs/2010.03768, 2020. URL https://api.semanticscholar.org/CorpusID:222208810.

Ruoyao Wang, Peter Alexander Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://api.semanticscholar.org/CorpusID:247451124.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *ArXiv*, abs/2505.22618, 2025. URL https://api.semanticscholar.org/CorpusID:278959508.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

A TASK CORRUPTION VERIFICATION

We evaluate task sensitivity on 208 retry states from ALFWorld benchmark. For each state s_i, we retain the interaction history and compare the repeated action a_i^{rep} with a distinct admissible alternative a_i^{alt}, selected by Qwen2.5-7B-Instruct. Both actions remain fixed while the original task $u_{i,0}$ is replaced by $K = 10$ randomly sampled task descriptions. Let $p_m(a \mid s, u)$ denote the action score for model m: summed autoregressive token log likelihood for Qwen, or a Monte Carlo diffusion variational score estimate for LLaDA. LLaDA uses identical sampled masks across task conditions.

For visualization, both actions are centered on the repeated action's original-task score:

$$D_{i,j}^{m,a} = p_m(a \mid s_i, u_{i,j}) - p_m(a_i^{\text{rep}} \mid s_i, u_{i,0}), \qquad a \in \{a^{\text{rep}}, a^{\text{alt}}\}. \tag{12}$$

The plotted curves are the state averages, $\bar{D}_j^{m,a} = N^{-1} \sum_{i=1}^{N} D_{i,j}^{m,a}$. This common baseline preserves the relative preference between the two actions; the alternative curve is therefore not centered on its own original-task score.

We quantify task sensitivity using the action-preference margin:

$$\Delta_{i,j}^{m} = p_m(a_i^{\text{alt}} \mid s_i, u_{i,j}) - p_m(a_i^{\text{rep}} \mid s_i, u_{i,j}). \tag{13}$$

For each state, we compute its population standard deviation across the replacement tasks, then average over states:

$$\sigma_m = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{1}{K} \sum_{j=1}^{K} \left(\Delta_{i,j}^{m} - \bar{\Delta}_i^{m}\right)^2}, \qquad \bar{\Delta}_i^{m} = \frac{1}{K} \sum_{j=1}^{K} \Delta_{i,j}^{m}. \tag{14}$$

The original task is excluded from this standard deviation. Smaller σ_m indicates less variation in action preference across replacement goals.

We additionally measure how frequently task replacement reverses the preferred action. Defining $b_{i,j}^{m} = \mathbf{1}[\Delta_{i,j}^{m} > 0]$, the preference flip rate is

$$F_m = \frac{1}{NK} \sum_{i=1}^{N} \sum_{j=1}^{K} \left| b_{i,j}^{m} - b_{i,0}^{m} \right|. \tag{15}$$

Each replacement is compared with the original task, yielding 2,080 comparisons per model. These calculations give $\sigma_{\text{LLaDA}} = 1.01$ and $\sigma_{\text{Qwen}} = 4.71$ nats, with flip rates of 6.6% and 44.4%, respectively.

The shaded bands use 2,000 episode-level bootstrap resamples to obtain pointwise 95% intervals for each action's change from its own original-task score. For the alternative curve, these intervals are shifted by the observed mean original-task preference margin to match the plotted baseline; they therefore represent uncertainty in the task-induced change, conditional on that baseline offset.

B REFLECT REVERSE ALGORITHM

We summarize the method introduced in Section 3.1 in the following pseudocode.

Algorithm 1 Reflect Reverse

Require: State s, task u, dLLM π_θ, number of proposals K, number of reflection rollouts M, inverse temperature β
Ensure: Selected action $a^\star$

Propose:
Sample K action candidates $a_1, \dots, a_K \sim \pi_\theta(\cdot \mid s, u)$ with ReAct
Deduplicate candidates to obtain $\mathcal{A}(s)$
Reflect:
for each $a \in \mathcal{A}(s)$ **do**
  $\phi(a) \leftarrow 0$
  for $m = 1, \dots, M$ **do**
    Sample a subset of task-token positions $B_m \subseteq \{1, \dots, |u|\}$
    Construct $\tilde{u}^{(m)}$ by masking positions B_m in u
    Condition the dLLM on $(s, a, \tilde{u}^{(m)})$
    $\ell_m(a) \leftarrow \dfrac{1}{|B_m|} \displaystyle\sum_{j \in B_m} \log \pi_\theta\big(u_j \mid s, a, \tilde{u}^{(m)}\big)$
    $\phi(a) \leftarrow \phi(a) + \ell_m(a)/M$
  end for
end for
Act:
$\rho(a \mid s, u) \leftarrow \dfrac{\exp(\beta\phi(a))}{\sum_{a' \in \mathcal{A}(s)} \exp(\beta\phi(a'))} \quad \forall a \in \mathcal{A}(s)$
Sample $a^\star \sim \rho(\cdot \mid s, u)$
return $a^\star$

C PROOF AND THEORY

Our method uses the model's reverse conditional $\pi_\theta(u \mid s, a)$, which masked dLLMs evaluate natively. We assume the two conditionals are consistent views of one underlying joint.

Assumption C.1 (Coherent conditionals)**.** *The model's forward and reverse conditionals satisfy Bayes' rule with respect to the task prior $p(u \mid s)$: for every a and every u in the support of $p(\cdot \mid s)$,*

$$\pi_\theta(u \mid s, a) \;=\; \frac{p(u \mid s)\, \pi_\theta(a \mid s, u)}{\sum_{u'} p(u' \mid s)\, \pi_\theta(a \mid s, u')}. \tag{16}$$

Assumption C.1 is an idealization of the same kind used whenever a single model's conditionals are composed; it is what licenses reading the reverse score as a posterior. A stronger variant, in which $\pi_\theta(u \mid s, a)$ is coherent with the task posterior $p(u \mid s, a)$ as well, removes the task prior from the proposition entirely (Remark C.3).

C.1 PROOF OF THE RECOVERY PROPOSITION

This appendix proves Proposition 3.1. The argument has two steps. The first step (Lemma C.1) isolates the only property of the task prior that matters: it can be changed arbitrarily without moving the induced task posterior, provided the corruption is re-absorbed accordingly. The second step substitutes Assumption 2.1 and lets the corruption cancel.

Lemma C.1 (Prior rescaling leaves the posterior untouched)**.** *Let $q(u \mid s)$ and $q'(u \mid s)$ be two task priors with the same support, and let $\nu(a \mid s, u)$ be any conditional. Define*

$$\nu_q(u \mid s, a) \;:=\; \frac{q(u \mid s)\, \nu(a \mid s, u)}{\sum_{u'} q(u' \mid s)\, \nu(a \mid s, u')}, \tag{17}$$

and $\nu_{q'}(u \mid s, a)$ analogously with q' in place of q. Then $\nu_q(u \mid s, a) = \nu_{q'}(u \mid s, a)$ for every a if $q'(u \mid s)/q(u \mid s)$ is constant in u on the support; in particular the induced posterior is unchanged whenever q' is replaced by q itself.

Proof. Immediate from Eq.17: replacing q by q' multiplies the numerator at u by $q'(u \mid s)/q(u \mid s)$ and the denominator by the same factor averaged over u'; the two coincide for every a exactly when the ratio does not depend on u. The trivial case $q' = q$ gives the invariance used below. □

Proposition (Proposition 3.1 restated; exact recovery of the faithful task posterior)**.** *Let Assumptions 2.1 and C.1 hold. Then, for every state s, task u in the support of $p(\cdot \mid s)$, and action a,*

$$\pi_\theta(u \mid s, a) \;=\; \frac{\omega(u \mid s)\, \mu(a \mid s, u)}{\sum_{u'} \omega(u' \mid s)\, \mu(a \mid s, u')}, \tag{18}$$

where

$$\omega(u \mid s) \;:=\; \frac{p(u \mid s)}{Z(s, u)} \tag{19}$$

is a positive task weight that does not depend on the candidate action a, and $Z(s, u) = \sum_{a'} \mu(a' \mid s, u)\, e^{\varepsilon(s,a')}$ is the normalizer of Assumption 2.1.

Proof. By Assumption C.1, the model's reverse conditional satisfies

$$\pi_\theta(u \mid s, a) \;=\; \frac{p(u \mid s)\, \pi_\theta(a \mid s, u)}{\sum_{u'} p(u' \mid s)\, \pi_\theta(a \mid s, u')}. \tag{20}$$

Apply Assumption 2.1 to every term. The numerator at u becomes

$$p(u \mid s)\, \pi_\theta(a \mid s, u) \;=\; p(u \mid s)\, \frac{\mu(a \mid s, u)\, e^{\varepsilon(s,a)}}{Z(s, u)}, \tag{21}$$

and the denominator becomes

$$\sum_{u'} p(u' \mid s)\, \pi_\theta(a \mid s, u') \;=\; \sum_{u'} p(u' \mid s)\, \frac{\mu(a \mid s, u')\, e^{\varepsilon(s,a)}}{Z(s, u')} \;=\; e^{\varepsilon(s,a)} \sum_{u'} \frac{p(u' \mid s)}{Z(s, u')}\, \mu(a \mid s, u'), \tag{22}$$

where the factor $e^{\varepsilon(s,a)}$ carries no task index u' and factors out of the sum. Substituting Eq.21 and Eq.22 into Eq.20, the factor $e^{\varepsilon(s,a)}$ appears once in the numerator and once in the denominator and cancels exactly:

$$\pi_\theta(u \mid s, a) \;=\; \frac{\big(p(u \mid s)/Z(s, u)\big)\, \mu(a \mid s, u)}{\sum_{u'} \big(p(u' \mid s)/Z(s, u')\big)\, \mu(a \mid s, u')} \;=\; \frac{\omega(u \mid s)\, \mu(a \mid s, u)}{\sum_{u'} \omega(u' \mid s)\, \mu(a \mid s, u')},$$

which is Eq.18. Lemma C.1 is what licenses reading the right-hand side as a task posterior: the prior has been rescaled from p to ω, but the induced posterior remains a genuine Bayes posterior of μ, simply under the prior ω. □

Identity Eq.18 holds for every action individually, so it preserves every feature of the comparison of actions: for any a_1, a_2,

$$\log \pi_\theta(u \mid s, a_1) - \log \pi_\theta(u \mid s, a_2) \;= \tag{23}$$

$$\log \frac{\mu(a_1 \mid s, u)}{\sum_{u'} \omega(u' \mid s)\, \mu(a_1 \mid s, u')} \;-\; \log \frac{\mu(a_2 \mid s, u)}{\sum_{u'} \omega(u' \mid s)\, \mu(a_2 \mid s, u')}, \tag{24}$$

with equal margins, equal ties, and hence the same ordering, top-k set, and argmax as the faithful model's task-evidence ratio; the margins are those of the ratio, and need not coincide with the margins of $\mu(\cdot \mid s, u)$ itself (Remark C.1).

Corollary C.1 (The corrected policy is the softmax of the faithful task-evidence ratio, at any temperature)**.** *Let $\mathcal{A}(s)$ be a finite candidate set and $\beta > 0$ an inverse temperature. Under Assumptions 2.1 and C.1,*

$$\frac{\pi_\theta(u \mid s,a)^\beta}{\sum_{a' \in \mathcal{A}(s)} \pi_\theta(u \mid s,a')^\beta} \;=\; \frac{\big(\mu(a \mid s,u) \big/ \sum_{u'} \omega(u' \mid s)\, \mu(a \mid s,u')\big)^\beta}{\sum_{a' \in \mathcal{A}(s)} \big(\mu(a' \mid s,u) \big/ \sum_{u'} \omega(u' \mid s)\, \mu(a' \mid s,u')\big)^\beta} \tag{25}$$

for every $a \in \mathcal{A}(s)$: an exact equality of probability distributions over candidates. The limit $\beta \to \infty$ recovers the argmax statement; $\beta \to 0$ recovers uniform sampling.

Proof. By Proposition 3.1, for each a the score $\pi_\theta(u \mid s,a)$ equals the faithful task-evidence ratio times the factor $\omega(u \mid s)$, which does not depend on a. Raising to β and normalizing over $\mathcal{A}(s)$ removes this shared factor from both sides. □

Remark C.1 (Ratio versus conditional)**.** *The right-hand side of Eq.25 is the softmax of the faithful task-evidence ratio $\mu(a \mid s,u) \big/ \sum_{u'} \omega(u' \mid s)\, \mu(a \mid s,u')$, not of the faithful conditional $\mu(a \mid s,u)$ itself: the task-marginal likelihood $m(a) = \sum_{u'} \omega(u' \mid s)\, \mu(a \mid s,u')$ depends on the candidate and is not removed by normalization. The two policies coincide when m is constant over $\mathcal{A}(s)$, e.g. when the candidates are equally plausible under the task mixture, or in the argmax limit $\beta \to \infty$, where only the ordering survives. Unconditionally, what is recovered exactly is the faithful task-evidence ratio's comparison of candidates: ordering and ties.*

Remark C.2 (On the weights ω)**.** *The faithful baseline on the right of Eq.18 mixes tasks with weights $\omega(u \mid s) = p(u \mid s)/Z(s,u)$ rather than with the prior $p(u \mid s)$ itself, because the corruption rescales each task-conditional by its own normalizer $Z(s,u)$. Only two properties of ω are used: it is positive and it does not depend on the candidate action a. If $Z(s,u)$ happens to be constant in u, i.e. the corruption rescales every task-conditional by the same total mass, then $\omega \propto p$ and the baseline is exactly the task-marginal of μ under the original prior.*

Remark C.3 (Dropping the task prior)**.** *If the model's conditionals are coherent with the task posterior as well, i.e. $\pi_\theta(u \mid s,a)$ directly proportional to $p(u \mid s,a)\,\pi_\theta(a \mid s,u)$, then the same cancellation applies to $p(u \mid s,a)$ itself, and Proposition 3.1 holds with $\omega(u \mid s)$ replaced by $p(u \mid s,a)/Z(s,u)$. In this case the method requires only the single reverse score $\pi_\theta(u \mid s,a)$: no explicit task prior and no baseline forward pass enter at all.*

C.2 Residual bias of the corrected policy

Remark C.1 shows that the corrected policy ρ is exactly the softmax of the faithful task-evidence ratio, and asks how far it remains from the faithful conditional itself. This appendix answers the question: it proves Theorem 3.1 of Section 3.3, via a sharp range lemma (Lemma C.2). Throughout, fix a state s, a task u in the support of $p(\cdot \mid s)$, and a finite candidate set $\mathcal{A} \subseteq \operatorname{supp} \mu(\cdot \mid s,u)$ with $|\mathcal{A}| \geq 2$; recall ρ from Eq.3, the positive task weights ω from Proposition 3.1, the task-marginal likelihood $m(a) = \sum_{u'} \omega(u' \mid s)\, \mu(a \mid s,u')$, and the restricted distributions and spreads of Eq.7.

Support identity. Since $e^{\varepsilon(s,a)} > 0$, Assumption 2.1 gives $\pi_\theta(a \mid s,u) > 0$ if and only if $\mu(a \mid s,u) > 0$. Hence a candidate set proposed from $\pi_\theta(\cdot \mid s,u)$ automatically satisfies $\mathcal{A} \subseteq \operatorname{supp} \mu(\cdot \mid s,u)$, and on $\mathcal{A}$ we have $\mu(a \mid s,u) > 0$ as well as $m(a) \geq \omega(u \mid s)\, \mu(a \mid s,u) > 0$. Moreover, by Eq.1,

$$\pi_{\mathcal{A}}(a) \;=\; \frac{\mu_{\mathcal{A}}(a)\, e^{\varepsilon(s,a)}}{\sum_{a' \in \mathcal{A}} \mu_{\mathcal{A}}(a')\, e^{\varepsilon(s,a')}}, \qquad a \in \mathcal{A}, \tag{26}$$

since the normalizer $Z(s,u)$ of Eq.1 cancels between numerator and denominator: the corrupted restriction is the faithful restriction tilted by the corruption.

A sharp range bound. Lemma C.2 below quantifies a simple fact: two distributions whose likelihood ratio is confined to a band cannot be far apart in total variation, and the worst case for a band of logarithmic width Δ is exactly $\tanh(\Delta/4)$. The key input is a sharp result of Binette (2019) (Corollary 5): if the likelihood ratio dP/dQ has essential infimum m and essential supremum M, then $d_{\mathrm{TV}}(P,Q) \leq \frac{(M-1)(1-m)}{M-m}$, with the best possible constant for fixed (m,M). Lemma C.2 is what this bound becomes under the scale-invariant constraint that only the *range* of $\log(dQ/dP)$, and not the location of the band, is known.

Lemma C.2 (Range bound, sharp)**.** *Let P and Q be strictly positive probability distributions on a finite set S, and suppose the log-likelihood ratio has range at most Δ:*

$$\max_{x\in S} \log \frac{Q(x)}{P(x)} \;-\; \min_{x\in S} \log \frac{Q(x)}{P(x)} \;\le\; \Delta. \tag{27}$$

Then

$$d_{\mathrm{TV}}(P,Q) \;\le\; \tanh\big(\Delta/4\big), \qquad \textit{where} \qquad d_{\mathrm{TV}}(P,Q) := \tfrac{1}{2}\sum_{x\in S} |P(x)-Q(x)|. \tag{28}$$

The constant is sharp: for every Δ there is a two-point family attaining equality.

Lemma C.2 is obtained by Corollary 5 of Binette (2019), followed by optimizing over the location of the bounded likelihood-ratio interval.

Proof of Theorem 3.1. (i) By Eq.25, $\rho(a) = R(a)/\sum_{a'\in\mathcal{A}} R(a')$ with $R(a) = (\mu(a \mid s,u)/m(a))^{\beta}$. Hence for every $a \in \mathcal{A}$,

$$\log \frac{\rho(a)}{\mu_{\mathcal{A}}(a)} = (\beta-1)\log \mu(a \mid s,u) - \beta \log m(a) + \underbrace{\Big(\log \mu(\mathcal{A} \mid s,u) - \log \textstyle\sum_{a'\in\mathcal{A}} R(a')\Big)}_{\text{constant in } a}. \tag{29}$$

The range of $\log(\rho/\mu_{\mathcal{A}})$ over $\mathcal{A}$ is therefore the range of $(\beta-1)\log\mu(\cdot \mid s,u) - \beta\log m$, which is at most $|\beta-1|\,\Lambda_\mu + \beta\Lambda_m$. Lemma C.2, applied to $P = \mu_{\mathcal{A}}$ and $Q = \rho$ (both strictly positive on $\mathcal{A}$ by the support identity), gives the first inequality in Eq.9.

For the second inequality we show $\Lambda_m \le \Lambda_{\mathrm{task}}$. Fix $a_1, a_2 \in \mathcal{A}$ and let $S_2 := \{u' \in \operatorname{supp} p(\cdot \mid s) : \mu(a_2 \mid s,u') > 0\}$. If some $u' \notin S_2$ has $\mu(a_1 \mid s,u') > 0$, then $\log\mu(a_1 \mid s,u') - \log\mu(a_2 \mid s,u') = +\infty$, so $\Lambda_{\mathrm{task}} = +\infty$ and the claim is vacuous. Otherwise $\mu(a_1 \mid s,u') = 0$ off S_2, and the mediant inequality (a weighted mean of ratios lies below the largest ratio) gives

$$\frac{m(a_1)}{m(a_2)} = \frac{\sum_{u'\in S_2} \omega(u' \mid s)\,\mu(a_2 \mid s,u')\,\big[\mu(a_1 \mid s,u')/\mu(a_2 \mid s,u')\big]}{\sum_{u'\in S_2} \omega(u' \mid s)\,\mu(a_2 \mid s,u')} \;\le\; \max_{u'\in S_2} \frac{\mu(a_1 \mid s,u')}{\mu(a_2 \mid s,u')},$$

so $\log m(a_1) - \log m(a_2) \le \max_{u'} \big[\log\mu(a_1 \mid s,u') - \log\mu(a_2 \mid s,u')\big] \le \Lambda_{\mathrm{task}}$. Taking the maximum over $a_1, a_2 \in \mathcal{A}$ yields $\Lambda_m \le \Lambda_{\mathrm{task}}$, and $\tanh$ is increasing.

(ii) By Eq.26, writing $\varepsilon(a) := \varepsilon(s,a)$ and $\varepsilon^+ := \varepsilon(a^+)$,

$$\pi_{\mathcal{A}}(a^+) = \frac{\gamma\, e^{\varepsilon^+}}{\sum_{a\in\mathcal{A}} \mu_{\mathcal{A}}(a) e^{\varepsilon(a)}} \;\ge\; \frac{\gamma\, e^{\varepsilon^+}}{\gamma e^{\varepsilon^+} + (1-\gamma)e^{\varepsilon^+ - \Delta_2}} = \frac{\gamma}{\gamma + (1-\gamma)e^{-\Delta_2}},$$

since $\varepsilon(a) \le \varepsilon^+ - \Delta_2$ for all $a \ne a^+$ by definition of Δ_2; equality holds when $|\mathcal{A}| = 2$, as there is only one other candidate. Moreover $\pi_{\mathcal{A}}(a^+) \ge \gamma = \mu_{\mathcal{A}}(a^+)$, because $\sum_a \mu_{\mathcal{A}}(a) e^{\varepsilon(a)} \le e^{\varepsilon^+}$. Taking the event $E = \{a^+\}$ in $d_{\mathrm{TV}}(\mu_{\mathcal{A}}, \pi_{\mathcal{A}}) = \sup_E\big(\pi_{\mathcal{A}}(E) - \mu_{\mathcal{A}}(E)\big)$,

$$d_{\mathrm{TV}}(\mu_{\mathcal{A}}, \pi_{\mathcal{A}}) \;\ge\; \pi_{\mathcal{A}}(a^+) - \gamma \;\ge\; \frac{\gamma}{\gamma + (1-\gamma)e^{-\Delta_2}} - \gamma = \frac{\gamma(1-\gamma)\big(1-e^{-\Delta_2}\big)}{\gamma + (1-\gamma)e^{-\Delta_2}}.$$

For the monotonicity claim, write $x = e^{-\Delta_2}$: the right-hand side is $\gamma(1-\gamma)(1-x)/(\gamma+(1-\gamma)x)$, with derivative $-\gamma(1-\gamma)/(\gamma+(1-\gamma)x)^2 < 0$ in x; hence it increases in Δ_2 and tends to $\gamma(1-\gamma)/\gamma = 1-\gamma$ as $x \to 0$.

□

Remark C.4 (Finite-M estimation enters additively)**.** *Theorem 3.1 analyzes the exact reverse score $\phi(a) = \log\pi_\theta(u \mid s,a)$, as does Proposition 3.1; the algorithm uses the Monte Carlo estimate $\hat\phi$ of Eq.2. If the estimation error $\delta(a) := \hat\phi(a) - \phi(a)$ satisfies $\sup_{a\in\mathcal{A}} |\delta(a)| \le \delta_{\max}$, then for the estimated policy $\hat\rho(a) \propto e^{\beta\hat\phi(a)}$,*

$$\log \frac{\hat\rho(a)}{\rho(a)} = \beta\,\delta(a) - \log \mathbb{E}_\rho\big[e^{\beta\delta}\big] \qquad \Longrightarrow \qquad d_{\mathrm{TV}}(\hat\rho, \rho) \le \tanh\big(\beta\delta_{\max}/2\big),$$

by Lemma C.2, since the range of $\beta\delta$ over $\mathcal{A}$ is at most $2\beta\delta_{\max}$. Hence $d_{\mathrm{TV}}(\hat\rho, \mu_{\mathcal{A}}) \le \tanh(\beta\delta_{\max}/2) + B$: estimation noise degrades the corruption-free guarantee gracefully and vanishes as the estimator becomes consistent.

D HYPERPARAMETERS

Table 2 summarizes the hyperparameters used for action generation and reflection. On ScienceWorld and Jericho, **Reflect Reverse** uses more Monte Carlo sampling steps than **Reflect Forward** because the task description is longer than the candidate actions at each interaction step, resulting in more tokens to evaluate during reverse reflection.

Due to iLLaDA's pre-training configuration, the assistant prefill and end-of-thinking sequence is `</think>` followed by a newline and `Thought:`, while the visible response begins with `Thought:` followed by a space.

Table 2: Evaluation hyperparameters for LLaDA and iLLaDA on ALFWorld, ScienceWorld, BabyAI, and Jericho. The upper table reports model settings and the lower table reports forward and reverse likelihood-scoring parameters

Hyperparameters	**LLaDA**	**iLLaDA**
Generation length (tokens)	128	128
Requested diffusion steps	128	128
Block size (tokens)	32	32
Threshold	0.9	0.9
Dual cache	On	Off
Configured context length (tokens)	4,000	8,192
Temperature	$\{0.8, 0.9\}$	$\{0.9, 1.0\}$
Seeds	$\{12, 22, 32\}$	$\{12, 22, 32\}$
Maximum agent steps per episode	30	30
Generation attempt limit	20	20

	Reflect Forward		**Reflect Reverse**	
Benchmark	**MC rollouts**	**Batch size**	**MC rollouts**	**Batch size**
ALFWorld	64	16	64	16
ScienceWorld	64	16	128	16
BabyAI	64	16	64	16
Jericho	64	16	128	16

E SUPPLEMENTARY EXPERIMENT RESULTS

We provide the full experimental results omitted from the main paper for space. Tables 3 and 4 report performance across all benchmarks, model variants, methods, and sampling temperatures, offering a more detailed view of the results summarized in the main text.

Table 3: Success rate (SR) and progress rate (PR) evaluated by LLaDA-8B-Instruct. Entries are mean ± sample standard deviation. Boldface marks the best method within each benchmark and temperature category (including Mean), with ties included.

Benchmark	Method	Temperature	SR (%)	PR (%)
ALFWorld	One Pass	0.8	6.47 ± 2.40	26.87 ± 1.46
		0.9	8.71 ± 1.55	29.33 ± 0.79
		Mean	7.59 ± 2.18	28.10 ± 1.71
	Reflect Forward	0.8	7.46 ± 1.49	29.73 ± 2.18
		0.9	8.96 ± 0.75	31.45 ± 1.02
		Mean	8.21 ± 1.33	30.59 ± 1.79
	Reflect Reverse	0.8	$\mathbf{12.19} \pm 2.28$	$\mathbf{34.29} \pm 3.03$
		0.9	$\mathbf{16.17} \pm 2.28$	$\mathbf{36.24} \pm 1.01$
		Mean	$\mathbf{14.18} \pm 2.99$	$\mathbf{35.26} \pm 2.29$
ScienceWorld	One Pass	0.8	3.70 ± 1.70	21.69 ± 4.42
		0.9	4.07 ± 0.64	$\mathbf{23.02} \pm 1.21$
		Mean	3.89 ± 1.17	22.36 ± 2.99
	Reflect Forward	0.8	3.70 ± 1.28	20.03 ± 2.37
		0.9	5.56 ± 1.92	22.26 ± 2.34
		Mean	4.63 ± 1.78	21.15 ± 2.43
	Reflect Reverse	0.8	$\mathbf{7.78} \pm 1.92$	$\mathbf{23.24} \pm 2.91$
		0.9	$\mathbf{5.93} \pm 1.28$	22.41 ± 1.26
		Mean	$\mathbf{6.85} \pm 1.78$	$\mathbf{22.82} \pm 2.06$
BabyAI	One Pass	0.8	14.88 ± 0.52	25.97 ± 0.66
		0.9	14.29 ± 0.89	27.46 ± 1.02
		Mean	14.58 ± 0.73	26.71 ± 1.12
	Reflect Forward	0.8	$\mathbf{18.75} \pm 4.72$	$\mathbf{30.41} \pm 3.19$
		0.9	16.37 ± 1.03	$\mathbf{28.39} \pm 0.77$
		Mean	$\mathbf{17.56} \pm 3.32$	$\mathbf{29.40} \pm 2.35$
	Reflect Reverse	0.8	17.86 ± 1.79	26.55 ± 1.17
		0.9	$\mathbf{17.26} \pm 3.61$	27.12 ± 1.72
		Mean	$\mathbf{17.56} \pm 2.57$	26.83 ± 1.35
Jericho	One Pass	0.8	$\mathbf{1.67} \pm 2.89$	19.59 ± 6.37
		0.9	0.00 ± 0.00	20.35 ± 3.17
		Mean	0.83 ± 2.04	19.97 ± 4.52
	Reflect Forward	0.8	0.00 ± 0.00	17.52 ± 2.50
		0.9	$\mathbf{3.33} \pm 2.89$	20.94 ± 1.39
		Mean	$\mathbf{1.67} \pm 2.58$	19.23 ± 2.61
	Reflect Reverse	0.8	0.00 ± 0.00	$\mathbf{20.53} \pm 2.22$
		0.9	$\mathbf{3.33} \pm 5.77$	$\mathbf{26.72} \pm 2.47$
		Mean	$\mathbf{1.67} \pm 4.08$	$\mathbf{23.63} \pm 3.99$

Table 4: Success rate (SR) and progress rate (PR) evaluated by iLLaDA-8B-Instruct. Entries are mean ± sample standard deviation. Boldface marks the best method within each benchmark and temperature category (including Mean), with ties included.

Benchmark	Method	Temperature	SR (%)	PR (%)
ALFWorld	One Pass	0.9	5.47 ± 0.43	19.57 ± 1.76
		1.0	4.73 ± 0.43	21.06 ± 1.81
		Mean	5.10 ± 0.56	20.32 ± 1.79
	Reflect Forward	0.9	0.75 ± 0.75	15.80 ± 1.39
		1.0	0.75 ± 1.29	16.36 ± 1.97
		Mean	0.75 ± 0.94	16.08 ± 1.56
	Reflect Reverse	0.9	$\mathbf{9.95} \pm 1.88$	$\mathbf{27.69} \pm 2.86$
		1.0	$\mathbf{10.20} \pm 1.14$	$\mathbf{28.71} \pm 1.55$
		Mean	$\mathbf{10.07} \pm 1.40$	$\mathbf{28.20} \pm 2.13$
ScienceWorld	One Pass	0.9	0.00 ± 0.00	15.77 ± 0.22
		1.0	0.37 ± 0.64	16.47 ± 1.73
		Mean	0.19 ± 0.45	16.12 ± 1.17
	Reflect Forward	0.9	0.37 ± 0.64	7.70 ± 1.21
		1.0	0.00 ± 0.00	9.82 ± 1.48
		Mean	0.19 ± 0.45	8.76 ± 1.68
	Reflect Reverse	0.9	$\mathbf{3.70} \pm 1.28$	$\mathbf{25.95} \pm 2.43$
		1.0	$\mathbf{4.07} \pm 2.31$	$\mathbf{25.39} \pm 0.84$
		Mean	$\mathbf{3.89} \pm 1.69$	$\mathbf{25.67} \pm 1.66$
BabyAI	One Pass	0.9	9.23 ± 2.25	22.65 ± 2.07
		1.0	11.61 ± 2.36	25.13 ± 2.82
		Mean	10.42 ± 2.44	23.89 ± 2.60
	Reflect Forward	0.9	8.93 ± 0.89	21.52 ± 1.15
		1.0	8.04 ± 1.55	18.44 ± 2.49
		Mean	8.48 ± 1.23	19.98 ± 2.42
	Reflect Reverse	0.9	$\mathbf{15.77} \pm 2.25$	$\mathbf{30.73} \pm 1.47$
		1.0	$\mathbf{20.24} \pm 1.36$	$\mathbf{35.40} \pm 0.36$
		Mean	$\mathbf{18.01} \pm 2.96$	$\mathbf{33.07} \pm 2.73$
Jericho	One Pass	0.9	$\mathbf{1.67} \pm 2.89$	11.58 ± 2.39
		1.0	$\mathbf{1.67} \pm 2.89$	11.62 ± 2.61
		Mean	$\mathbf{1.67} \pm 2.58$	11.60 ± 2.24
	Reflect Forward	0.9	0.00 ± 0.00	9.63 ± 1.31
		1.0	0.00 ± 0.00	11.51 ± 1.08
		Mean	0.00 ± 0.00	10.57 ± 1.49
	Reflect Reverse	0.9	0.00 ± 0.00	$\mathbf{11.93} \pm 1.25$
		1.0	$\mathbf{1.67} \pm 2.89$	$\mathbf{21.37} \pm 2.69$
		Mean	0.83 ± 2.04	$\mathbf{16.65} \pm 5.50$

F RETRY STATE EXAMPLES

Tables 5 and 6 present two representative retry-state patterns from ALFWorld: single-action repetition and two-action alternation. Similar retry patterns are also frequently observed across other multi-turn benchmarks.

Table 5: Single-action example. A and O denote the action and its resulting observation. The repetition loop is marked in red including the initial action of the loop.

Look at pencil under the desklamp.
A: `go to desk 1` **O:** On the desk 1, you see a bowl 2, a creditcard 2, a desklamp 1, a laptop 1, a mug 1, and a pen 2.
A: `examine desk 1` **O:** On the desk 1, you see a bowl 2, a creditcard 2, a desklamp 1, a laptop 1, a mug 1, and a pen 2.
A: `go to desk 2` **O:** On the desk 2, you see a alarmclock 1, a bowl 3, and a pencil 1.
A: `examine desk 2` **O:** On the desk 2, you see a alarmclock 1, a bowl 3, and a pencil 1.
A: `examine desk 2` **O:** On the desk 2, you see a alarmclock 1, a bowl 3, and a pencil 1.
A: `examine desk 2` **O:** On the desk 2, you see a alarmclock 1, a bowl 3, and a pencil 1.
A: `examine desk 2` **O:** On the desk 2, you see a alarmclock 1, a bowl 3, and a pencil 1.
A: `examine desk 2` **O:** On the desk 2, you see a alarmclock 1, a bowl 3, and a pencil 1.
A: `examine desk 2` **O:** On the desk 2, you see a alarmclock 1, a bowl 3, and a pencil 1.
A: `take pencil 1 from desk 2` **O:** You pick up the pencil 1 from the desk 2.
A: `go to desk 1` **O:** On the desk 1, you see a bowl 2, a creditcard 2, a desklamp 1, a laptop 1, a mug 1, and a pen 2.
A: `use desklamp 1` **O:** You turn on the desklamp 1.

Table 6: Alternating-action example. A and O denote the action and its resulting observation. The repetition loop is marked in red including the pair of the loop.

Put a clean egg in microwave.
A: `go to fridge 1` **O:** The fridge 1 is closed.

Continued on next page

Put a clean egg in microwave. *(continued)*

A: `open fridge 1`
O: You open the fridge 1. The fridge 1 is open. In it, you see a apple 2, a egg 1, a lettuce 1, a pan 2, a plate 1, and a tomato 1.

A: `take egg 1 from fridge 1`
O: You pick up the egg 1 from the fridge 1.

A: `put egg 1 in/on fridge 1`
O: You put the egg 1 in/on the fridge 1.

A: `close fridge 1`
O: You close the fridge 1.

A: `open fridge 1`
O: You open the fridge 1. The fridge 1 is open. In it, you see a apple 2, a egg 1, a lettuce 1, a pan 2, a plate 1, and a tomato 1.

A: `close fridge 1`
O: You close the fridge 1.

A: `open fridge 1`
O: You open the fridge 1. The fridge 1 is open. In it, you see a apple 2, a egg 1, a lettuce 1, a pan 2, a plate 1, and a tomato 1.

A: `close fridge 1`
O: You close the fridge 1.

A: `open fridge 1`
O: You open the fridge 1. The fridge 1 is open. In it, you see a apple 2, a egg 1, a lettuce 1, a pan 2, a plate 1, and a tomato 1.

A: `take egg 1 from fridge 1`
O: You pick up the egg 1 from the fridge 1.

A: `put egg 1 in/on fridge 1`
O: You put the egg 1 in/on the fridge 1.

A: `go to microwave 1`
O: The microwave 1 is closed.

A: `open microwave 1`
O: You open the microwave 1. The microwave 1 is open. In it, you see a apple 1, and a potato 1.

A: `take apple 1 from microwave 1`
O: You pick up the apple 1 from the microwave 1.

A: `go to fridge 1`
O: The fridge 1 is open. In it, you see a apple 2, a egg 1, a lettuce 1, a pan 2, a plate 1, and a tomato 1.

A: `put apple 1 in/on fridge 1`
O: You put the apple 1 in/on the fridge 1.

A: `take egg 1 from fridge 1`
O: You pick up the egg 1 from the fridge 1.

A: `put egg 1 in/on fridge 1`
O: You put the egg 1 in/on the fridge 1.

Continued on next page

Put a clean egg in microwave. *(continued)*

A: `go to microwave 1`
O: The microwave 1 is open. In it, you see a potato 1.

A: `take potato 1 from microwave 1`
O: You pick up the potato 1 from the microwave 1.

A: `put potato 1 in/on microwave 1`
O: You put the potato 1 in/on the microwave 1.

A: `examine microwave 1`
O: The microwave 1 is open. In it, you see a potato 1.

A: `go to fridge 1`
O: The fridge 1 is open. In it, you see a apple 2, a apple 1, a egg 1, a lettuce 1, a pan 2, a plate 1, and a tomato 1.

A: `take egg 1 from fridge 1`
O: You pick up the egg 1 from the fridge 1.

A: `put egg 1 in/on fridge 1`
O: You put the egg 1 in/on the fridge 1.

A: `go to microwave 1`
O: The microwave 1 is open. In it, you see a potato 1.

A: `close microwave 1`
O: You close the microwave 1.

A: `go to fridge 1`
O: The fridge 1 is open. In it, you see a apple 2, a apple 1, a egg 1, a lettuce 1, a pan 2, a plate 1, and a tomato 1.

A: `take egg 1 from fridge 1`
O: You pick up the egg 1 from the fridge 1.