COLDSPLASH: A BORROWED THOUGHT FOR AGENT SELF-REFLECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

As language agents perform increasingly long and complex multi-turn tasks, robust performance requires not only correct initial actions but also the ability to recover from errors through re-planning. Self-reflection is a common mechanism for recovery, but agents can remain anchored to the interpretation that caused the failure, leading them to reproduce the same erroneous reasoning despite having access to relevant evidence in the trajectory. We introduce COLDSPLASH, a lightweight cross-model intervention that mitigates this anchoring effect by transferring only the opening of a reflection from a heterogeneous model. The original agent then completes the reflection and resumes the task using the same trajectory and evidence. Thus, COLDSPLASH changes the initialization of the reflection process without delegating the task or modifying the agent's subsequent reasoning process. We evaluate COLDSPLASH on the self-reflection failure sets of τ^2-bench across Airline, Retail, and Telecom. On the pooled Airline/Retail cohort, a 16-token opening improves pass@any(4), the fraction of tasks solved in at least one of four attempts, from 5.2% to 9.4%, an 80% relative improvement, with gains observed across actors and domains. Diagnostic audits, trajectory-level analyses, and reflection-channel analyses further show that the intervention can change subsequent behavior rather than merely the linguistic form of reflection. These results demonstrate that a small, well-placed external nudge can unlock recovery capacity that agents already latently possess, pointing to lightweight cross-model reflection as a practical complement to scaling a single model's introspection. Code and data is https://anonymous.4open.science/r/coldflash-1D84.

1 INTRODUCTION

An agent must not only act, but also decide when its interpretation of a task deserves another look. In a tool-use conversation, user constraints, tool observations, and the agent's own intermediate conclusions arrive at different times. Reasoning and interaction can support one another, yet an early assumption can also persist as the conversation develops (Shinn et al., 2023; Madaan et al., 2023; Huang et al., 2024; Kamoi et al., 2024; Gou et al., 2024; Kumar et al., 2025). A natural response is to ask the agent to reflect before continuing. Self-feedback and reflective memory have become useful components of language-model systems (Park et al., 2023; Zhao et al., 2024; Packer et al., 2023; Tan et al., 2025; Chhikara et al., 2025; Renze & Guven, 2024).

This introduces a second inference problem: *who reviews the review?* The agent must distinguish contradictions from missing information and propose a justified next step(Sun et al., 2024; Lan et al., 2024; Hu et al., 2024). A review can elaborate an unsupported endorsement or manufacture a rationale for criticism; neither opening is inherently correct. Prior limitations of intrinsic self-correction motivate examining this process, rather than assuming that requesting reflection suffices(Zhang et al., 2025a; Jiang et al., 2026; Xie et al., 2025).

We investigate a narrow intervention: *borrow the beginning of a reflection, not the responsibility for solving the task*. In COLDSPLASH, an auxiliary model B provides the first k reference-tokenizer tokens of a review note. The original actor A completes that note and then resumes acting. Both models see the same permitted history. There is no retrospective failure label, hidden answer, new tool observation, or weight update in the intervention. Our default $k = 16$ changes who initializes

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.

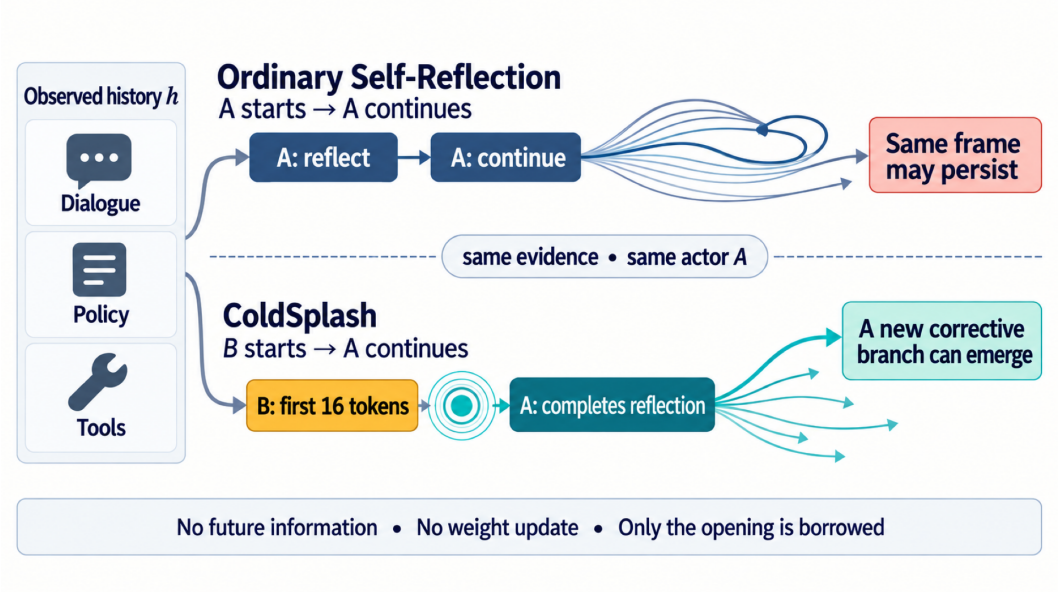

Figure 1: **A borrowed beginning, an original-model continuation.** At the same permitted history h, B supplies a short reflection opening z; A completes the review and resumes tool use. The proposed private-note channel does not elicit a user turn. The archived fork experiment instead exposes the note to the user simulator; that variant is evaluated separately.

the reflection while leaving most of its generation to A(Liu et al., 2025; Zhang et al., 2025b; Sun et al., 2026; Wang et al., 2024).

Our cognitive description is *reflective framing*: a review's opening may influence the interpretation it develops. This is an operational hypothesis about conditional text and action distributions, not a measurement of internal belief. The question is whether another opening helps A act more effectively, not simply produce more diverse text(Parcalabescu & Frank, 2024; Paul et al., 2024; Bentham et al., 2024; Tutek et al., 2025).

Our results show that changing the beginning of a reflection can improve downstream recovery without replacing or updating the original actor. We observe this effect across multiple actors and tool-use domains on selected failure cohorts, with public-note and private-note protocols evaluated separately.

Our contributions are threefold:

- **Observation.** We identify reflection initialization as an important factor in agent recovery: revisiting an interaction history does not guarantee a useful change in subsequent behavior. We theoretically characterize when shifting the distribution of reflection openings can redirect an unchanged actor toward more effective continuations.
- **Method.** We propose ColdSplash, a lightweight self-reflection framework that borrows only a short reflection opening from a heterogeneous model, allowing the original agent to complete the review and resume acting without weight updates or task delegation.
- **Empirical Findings.** We evaluate ColdSplash across three tool-use domains with multiple actor and helper models. A 16-token borrowed opening improves pooled Airline/Retail four-attempt recovery by 80.0%, 12.5%, and 50.0% relative across three actors, with additional gains in Telecom and a separate private-note evaluation. These results establish reflection initialization as a practical intervention for improving recovery in the evaluated settings.

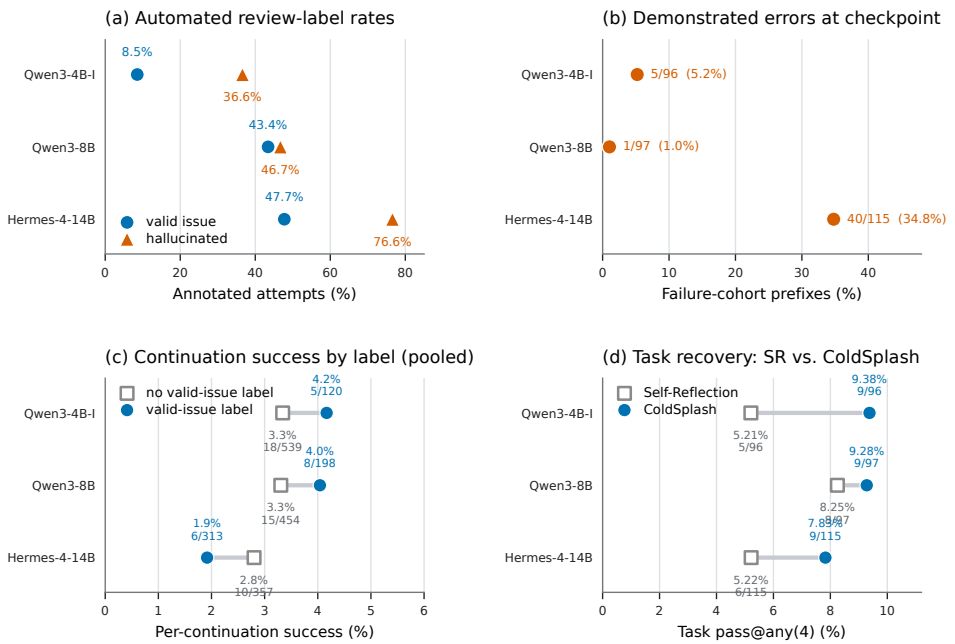

Figure 2: **Descriptive motivation audit.** Actors are Qwen3-4B-I, Qwen3-8B, and Hermes-4-14B, respectively. **(a)** Self-Reflection label rates (328/334/333 attempts from 89/92/87 tasks); valid-issue and hallucination labels may co-occur. **(b)** Prefixes labeled as containing checkpoint-visible errors ($N = 96/97/115$). **(c)** Continuation success by valid-issue label, pooling both methods ($N = 659/652/670$), without adjustment for within-task dependence. **(d)** Task-level pass@any(4), Self-Reflection versus COLDSPLASH: $5/96$ vs. $9/96$, $8/97$ vs. $9/97$, and $6/115$ vs. $9/115$. Automated labels are not human-validated ground truth; results establish neither causality nor statistical significance.

2 MOTIVATION

Self-reflection allows language agents to reconsider previous decisions and adjust their subsequent actions during multi-turn interactions (Madaan et al., 2023; Shinn et al., 2023). However, reflection is itself a model-generated decision: an agent must determine whether its previous actions were incorrect, identify relevant evidence, and propose an effective correction. Revisiting the same interaction history does not necessarily guarantee a more accurate interpretation or a more productive continuation. We therefore investigate two potential limitations of self-reflection and examine whether changing the starting point of a review can improve downstream recovery.

Failure mode 1: self-reflection can introduce unsupported criticism. We first examine whether an agent's self-review reliably identifies mistakes supported by the available interaction history. Using archived trajectories from the Airline and Retail domains of τ^2-Bench, we analyze self-reflection generated by three actors: Qwen3-4B-Instruct-2507, Qwen3-8B, and Hermes-4-14B. An external-model evaluator annotates whether each review identifies a valid issue or introduces unsupported claims, while separately assessing whether the interaction history contains an error demonstrable from the evidence available at the reflection checkpoint.

As shown in Figure 2(a), self-reflection can introduce unsupported criticism even when it identifies valid issues. Moreover, Figure 2(b) shows that only a subset of the failed trajectories contains errors identifiable from the checkpoint-visible evidence. This distinction is important: eventual task failure does not necessarily imply that the agent has already made an identifiable mistake at the time of reflection. Consequently, encouraging an agent to reconsider its previous decisions may introduce new errors when the available evidence does not justify a correction.

Failure mode 2: valid criticism does not guarantee recovery. Even when a reflection correctly identifies a problem, the agent must still translate that diagnosis into effective subsequent actions. To examine this distinction, we group archived continuations according to whether their preceding reviews identify a valid issue and compare their downstream task success rates.

Figure 2(c) shows that identifying a valid issue does not consistently correspond to higher task success across the three actors. A review may correctly criticize an earlier decision without leading to an effective correction, while a productive continuation need not begin with explicit criticism. These observations suggest that reflection quality alone is insufficient to characterize recovery: the relevant question is whether the review leads the agent toward more effective subsequent behavior.

Borrowed openings as an alternative starting point. The preceding observations motivate changing how a reflection begins, rather than simply requesting more criticism. We investigate a minimal intervention in which an auxiliary model B supplies only a short reflection opening, after which the original actor A completes the review and resumes the task. Both models receive the same available interaction history, and the auxiliary model does not provide a complete solution or execute subsequent actions.

To examine whether such an intervention can improve recovery, we compare fresh self-reflection with a borrowed reflection opening at the same fixed interaction checkpoint. We evaluate tasks that previously failed under self-reflection and measure recovery using pass@any(4), the fraction of tasks successfully completed in at least one of four continuation attempts.

As shown in Figure 2(d), borrowed openings improve observed four-attempt recovery over fresh self-reflection across all three actors. These results suggest that changing the initial context of reflection can provide access to more productive continuations without replacing or updating the original actor. However, improved task outcomes alone do not establish whether the intervention operates through more accurate diagnosis or other changes in the subsequent interaction.

Design implication. Together, these findings motivate a lightweight intervention at the beginning of reflection. Rather than relying on the agent to repeatedly reconsider its previous decisions from the same starting point, we introduce an alternative opening from a heterogeneous model and allow the original actor to complete the review and determine its subsequent actions. This motivates COLDSPLASH, which borrows only a brief reflection opening while preserving the original agent's responsibility for interpreting the available evidence and completing the task.

3 METHOD

We propose COLDSPLASH, a lightweight inference-time framework that improves agent recovery by changing how self-reflection is initialized, as illustrated in Figure 1. Rather than delegating reasoning or task execution to an auxiliary model, COLDSPLASH borrows only a short reflection opening from a heterogeneous model and lets the original actor complete the review and resume the task. The method consists of five steps: fixed-checkpoint intervention, reflection context construction, auxiliary opening generation, actor-guided reflection completion, and task continuation.

Step 1: Fixed-checkpoint intervention. Given a multi-turn task x, the original actor A interacts with the environment using its available tools. After a predefined number of completed tool transactions, we identify an intervention checkpoint and denote the interaction history available at this point by h. The checkpoint is fixed independently of the eventual task outcome, rather than retrospectively selected based on an observed error. Both ordinary self-reflection and COLDSPLASH operate on the same history h, without access to future observations or success labels.

Step 2: Reflection context construction. At the intervention checkpoint, we construct a review context c from the available interaction history, environment policy, and tool schemas:

$$c = \mathcal{P}(h, \text{policy}, \text{tools}),$$

where $\mathcal{P}$ denotes the reflection prompt and its input construction.

Under ordinary self-reflection, the actor generates a complete review:

$$r_{\text{SR}} \sim \pi_A(\cdot \mid c).$$

The resulting review is then used to condition subsequent task execution. However, this procedure initializes the reflection using the same model that generated the preceding interaction. We investigate whether an alternative initialization can redirect the actor toward a more productive continuation without changing the available evidence or its underlying model parameters.

Step 3: Auxiliary reflection opening generation. Instead of asking the original actor to initialize its own review, we let a heterogeneous auxiliary model B generate a candidate reflection under the same context:

$$u_B \sim \pi_B(\cdot \mid c).$$

We retain only its first k tokens under a fixed reference tokenizer T_*:

$$z_B = \mathrm{prefix}_{k,T_*}(u_B).$$

In our main configuration, $k = 16$. The auxiliary output is decoded into text and truncated using the reference tokenizer before being provided to the actor. This short textual opening serves as an alternative initialization for self-reflection, rather than a complete external review or an intermediate task solution.

Importantly, the auxiliary model receives no future trajectory, task outcome, or additional tool observation. It neither executes environment actions nor replaces the original actor's decision-making process.

Step 4: Actor-guided reflection completion. We provide the borrowed opening z_B to the original actor as an assistant prefill. The actor then generates the remaining reflection conditioned on the same review context:

$$s_A \sim \pi_A(\cdot \mid c, \text{ assistant-prefill} = z_B).$$

The completed reflection is

$$r_{\mathrm{CS}} = z_B \| s_A, \tag{1}$$

where $\|$ denotes textual concatenation.

This construction changes only the source of the reflection's opening. The original actor remains responsible for completing the review, interpreting the available evidence, and determining its subsequent actions. The borrowed opening may introduce an alternative question, concern, or proposed check, but its content is not treated as a verified observation or an instruction that overrides the existing task constraints.

Unlike full external reflection, COLDSPLASH does not adopt the auxiliary model's complete diagnosis or proposed solution. Instead, it changes the conditional starting point of the actor's own reflection while keeping both models' parameters frozen.

Step 5: Task continuation and recovery. After the reflection is completed, the original actor resumes task execution using the augmented interaction history. In the main fork-recovery implementation, the completed review is appended as an ordinary assistant message:

$$h' = h \| \,\mathrm{AssistantMessage}(r_{\mathrm{CS}}).$$

The actor then continues interacting with the environment using its original policy and tools:

$$y \sim \mathcal{T}_A(\cdot \mid h'),$$

where $\mathcal{T}_A$ denotes the actor's continuation under the environment and user-interaction protocol.

In this public-note protocol, the inserted review is visible to the user simulator, which may respond before the actor's next action. The resulting continuation therefore captures an interaction-level intervention rather than an isolated change to the actor's private reflection.

To evaluate recovery, we generate K attempts from the same intervention checkpoint. For each attempt j, a fresh auxiliary opening $z_B^{(j)}$ is generated and independently completed by the original

actor:

$$
\begin{aligned}
z_B^{(j)} &= \mathrm{prefix}_{k,T_*}\left(u_B^{(j)}\right), \quad u_B^{(j)} \sim \pi_B(\cdot \mid c), \\
s_A^{(j)} &\sim \pi_A(\cdot \mid c, \text{ assistant-prefill} = z_B^{(j)}), \\
r_{\mathrm{CS}}^{(j)} &= z_B^{(j)} \| s_A^{(j)}, \\
y^{(j)} &\sim \mathcal{T}_A\left(\cdot \mid h \| \,\mathrm{AssistantMessage}(r_{\mathrm{CS}}^{(j)})\right).
\end{aligned}
$$

In our main experiments, $K = 4$. Each attempt therefore receives a freshly initialized reflection, rather than sharing a single fixed review across multiple continuations. We compare this procedure against fresh self-reflection at the same checkpoint and evaluate recovery using pass@any(K).

Private-note variant. We additionally evaluate an isolated reflection channel in which the completed review is inserted only into the actor's constructed context rather than appended to the environment-visible conversation. Specifically, after two completed tool transactions, the actor receives the borrowed opening, completes its reflection, and resumes the task with the resulting note included only in its private view.

Unlike the main public-note implementation, the user simulator does not observe the reflection. This variant evaluates the effect of reflection initialization without directly exposing the generated note to the user simulator. Because the public-note and private-note implementations use different intervention checkpoints and interaction protocols, we report their results separately.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and models. We evaluate COLDSPLASH on the Airline, Retail, and Telecom domains of τ^2-bench. Our primary experiments use Qwen3-4B-Instruct-2507 (Qwen3-4B-I), Qwen3-8B, Hermes-3-8B, and Hermes-4-14B as actors. Hermes-3-Llama-3.2-3B is used as the default auxiliary model, including all Telecom experiments. Some actor checkpoints use different aliases across the two serving environments; we normalize these naming differences when reporting results.

Evaluation protocol. All three domains follow the same fixed-prefix recovery protocol. For each actor, we first construct an actor-specific Self-Reflection (SR) failure cohort. We then branch from the same interaction prefix after one completed tool transaction and compare fresh Self-Reflection with COLDSPLASH under the same continuation budget. Both methods regenerate the review for every attempt; COLDSPLASH differs by initializing the reflection with a fresh 16-token auxiliary opening, after which the original actor completes the review.

The completed review is inserted as a public assistant message before the actor resumes the task, so the user simulator may respond to the review before the actor's next action. We generate four attempts per task, using temperature 0.7, top-$p = 0.8$, a 20-step episode budget, and a 256-token limit per review-generation call. Further implementation and cohort-construction details are provided in Appendix A.

Metric. Our primary metric is pass@any(4), the fraction of tasks successfully completed in at least one of four attempts:

$$
S_4 = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\max_{1 \le j \le 4} R(y_i^{(j)}) = 1\right], \tag{2}
$$

where N is the number of evaluated tasks and R is the benchmark success reward. We describe improvement over Self-Reflection using the relative gain

$$
\Delta_{\mathrm{rel}} = 100 \, \frac{S_{4,\mathrm{CS}} - S_{4,\mathrm{SR}}}{S_{4,\mathrm{SR}}}\%. \tag{3}
$$

Domain-level results for the separate private-note evaluation are reported in Appendix D.

Table 1: **Recovery across actors and domains.** Pass@any(4) (%) on actor-specific SR-failure cohorts. All experiments follow the same fork@1 recovery protocol and use Hermes-3-Llama-3.2-3B as the auxiliary model. Rel. Gain denotes the relative improvement of COLDSPLASH over Self-Reflection.

Actor	Domain	N	SR	COLDSPLASH	Rel. Gain
Qwen3-4B-I	Airline/Retail	96	5.21%	**9.38%**	**+80.0%**
Qwen3-8B	Airline/Retail	97	8.25%	**9.28%**	**+12.5%**
Hermes-4-14B	Airline/Retail	115	5.22%	**7.83%**	**+50.0%**
Qwen3-4B-I	Telecom	26	7.69%	**23.08%**	**+200.0%**
Qwen3-8B	Telecom	20	15.00%	**20.00%**	**+33.3%**
Hermes-4-14B	Telecom	19	26.32%	**31.58%**	**+20.0%**

4.2 MAIN RESULTS

Recovery across actors and domains. We first ask whether changing the initialization of reflection improves downstream recovery. Table 1 compares COLDSPLASH with fresh Self-Reflection under the same intervention opportunity.

On the pooled Airline/Retail failure cohorts, COLDSPLASH increases pass@any(4) from 5.21% to 9.38% for Qwen3-4B-I, from 8.25% to 9.28% for Qwen3-8B, and from 5.22% to 7.83% for Hermes-4-14B, corresponding to relative gains of **80.0%, 12.5%, and 50.0%**, respectively.

The same pattern extends to Telecom. With the same 3B auxiliary model, Qwen3-4B-I improves from 7.69% to 23.08% (**+200.0% relative**), Qwen3-8B from 15.00% to 20.00% (**+33.3% relative**), and Hermes-4-14B from 26.32% to 31.58% (**+20.0% relative**), while Hermes-3-8B remains unchanged at 4.55%. Thus, the observed recovery benefit is not restricted to a single actor or domain, although its magnitude varies across configurations.

Interpretation. These results show that changing the opening of reflection can create additional recovery opportunities while leaving the original actor responsible for completing the review and executing subsequent actions. The effect is not uniform across actors or domains, which is consistent with our view of the borrowed opening as an alternative reflection initialization rather than a guaranteed correction. Because the main protocol exposes the completed review to the user simulator, these results characterize the full review–user–actor interaction.

4.3 INFERENCE EFFICIENCY

Reflection-generation latency. We next examine whether borrowing a short reflection opening introduces additional inference overhead. For this analysis, we use an optimized implementation, COLDSPLASH-opt, in which the auxiliary model generates only the retained 16-token opening before the actor continues the review. Table 2 compares its reflection-generation latency with ordinary Self-Reflection over 40 runs.

Across all three actors, COLDSPLASH-opt does not increase reflection latency. Mean latency decreases by **6.0%** for Qwen3-4B-I, **73.2%** for Qwen3-8B, and **35.9%** for Hermes-4-14B. The auxiliary seed itself requires only about 182–188 ms and accounts for **8.2%–19.8%** of total COLDSPLASH-opt reflection latency; the remaining **80.2%–91.8%** is spent on the original actor's continuation. Thus, the heterogeneous component remains a small fraction of the reflection-generation cost, while most generation is still performed by the original actor.

4.4 ABLATION STUDY

Does recovery require a stronger auxiliary model? We next vary the auxiliary model while keeping the actor and intervention structure fixed. We compare Hermes-3-Llama-3.2-3B, Hermes-3-Llama-3.1-8B, and Hermes-4-14B on fixed Airline/Retail recovery cohorts for Qwen3-4B-I and Qwen3-8B. All helpers provide the same 16-token reflection opening before the original actor completes the review.

Table 2: **Reflection-generation efficiency.** Mean ± standard deviation over 40 runs. SR is ordinary Self-Reflection; COLDSPLASH-opt uses a 16-token auxiliary opening followed by actor completion. B and A report auxiliary-seed and actor-continuation latency, with their mean share of total COLDSPLASH-opt latency in parentheses.

Actor	SR (ms)	COLDSPLASH-opt (ms)	Change	B (ms)	A (ms)
Qwen3-4B-I	2455 ± 573	2307 ± 446	-6.0%	182 ± 22 (8.2%)	2125 ± 446 (91.8%)
Qwen3-8B	5375 ± 126	1442 ± 670	-73.2%	188 ± 13 (19.8%)	1254 ± 663 (80.2%)
Hermes-4-14B	2337 ± 663	1498 ± 461	-35.9%	182 ± 16 (13.6%)	1316 ± 461 (86.4%)

Table 3: **Effect of auxiliary-model choice on recovery.** Pass@any(4) (%) on fixed Airline/Retail recovery cohorts. All auxiliary models provide the same 16-token reflection opening. Rel. Gain is measured relative to Self-Reflection for the same actor.

Actor	Reflection source	Pass@any(4)	Rel. Gain
Qwen3-4B-I	Self-Reflection	5.21%	–
Qwen3-4B-I	Hermes-3-3B	9.38%	**+80.0%**
Qwen3-4B-I	Hermes-3-8B	10.42%	**+100.0%**
Qwen3-4B-I	Hermes-4-14B	10.42%	**+100.0%**
Qwen3-8B	Self-Reflection	8.25%	–
Qwen3-8B	Hermes-3-3B	9.28%	**+12.5%**
Qwen3-8B	Hermes-3-8B	9.28%	**+12.5%**
Qwen3-8B	Hermes-4-14B	17.52%	**+112.4%**

As shown in Table 3, for Qwen3-4B-I, the 3B helper already increases pass@any(4) from 5.21% to 9.38%, corresponding to an **80.0% relative gain**. The 8B and 14B helpers both reach 10.42%, corresponding to **100.0% relative gains**.

For Qwen3-8B, the 3B and 8B helpers both increase recovery from 8.25% to 9.28% (**+12.5% relative**), whereas the 14B helper reaches 17.52% (**+112.4% relative**).

These results show that useful reflection initialization does not require the largest tested auxiliary model: even the 3B helper improves recovery for both actors. At the same time, the larger gain from the 14B helper for Qwen3-8B shows that auxiliary-model choice can affect the magnitude of the benefit. Because these checkpoints also differ in model family and generation, the comparison should not be interpreted as a controlled scaling law with respect to model size alone.

4.5 ROBUSTNESS ANALYSIS

Random-seed robustness. To examine sensitivity to sampling randomness, we repeat the Self-Reflection and COLDSPLASH comparison over three random seeds while keeping the actor, auxiliary model, task cohort, and intervention protocol fixed. As shown in Table 4, COLDSPLASH improves pass@any(4) under all three evaluated seeds, with relative gains ranging from 71.4% to 120.0%. The table reports mean recovery of 5.90% for Self-Reflection and 11.1% for COLDSPLASH, with an average seed-wise relative gain of 90.5%. The direction of the observed improvement is consistent across these three seeds.

Private-note evaluation. We additionally evaluate COLDSPLASH under a private-note protocol, where the completed reflection is visible only to the actor and not to the user simulator. For Qwen3-4B-I, pooled Airline/Retail pass@any(4) increases from 19.01% to 19.83%, corresponding to a **+4.3% relative improvement**. This evaluation differs from the main public-note experiment in intervention checkpoint, review sampling, generation limit, and episode budget. We therefore report it as a separate private-note evaluation, not as a controlled ablation of note visibility. Its effect size should not be directly compared with that of the public-note protocol. Complete private-note results are provided in Appendix D.

Table 4: **Robustness to sampling randomness.** Pass@any(4) (%) under different random seeds with a fixed actor-helper configuration and evaluation cohort. Rel. Gain denotes the relative improvement of ColdSplash over Self-Reflection.

Seed	N	SR	ColdSplash	Rel. Gain
101	96	5.21%	**9.38%**	**+80.0%**
202	96	7.29%	**12.50%**	**+71.4%**
303	96	5.21%	**11.46%**	**+120.0%**
Mean $\pm$ Std	96	$5.90 \pm 0.98\%$	$\mathbf{11.1 \pm 1.30\%}$	$\mathbf{+90.5 \pm 21.1\%}$

5 Related Work

Self-reflection and self-correction. Language-model agents use reflection to revise intermediate decisions, refine outputs, or accumulate experience across attempts (Madaan et al., 2023; Shinn et al., 2023) . Recent work further improves reflection through learned critique and failure-centric training, including online recovery from erroneous trajectories and multi-level retrospective reflection(Yuan et al., 2025; Ge et al., 2025). These approaches primarily change how reflections are generated, trained, or reused. ColdSplash studies a complementary design axis: the initialization of a single within-episode review. Rather than replacing the actor with a trained critic or storing a complete external reflection, a heterogeneous model supplies only a short opening, after which the original actor completes the review and retains responsibility for subsequent actions.

Critique quality and feedback utilization. A separate line of work asks whether models can reliably identify their own errors and make effective use of corrective feedback (Huang et al., 2024; Tyen et al., 2024; Yang et al., 2025). These studies show that critique generation and correction are distinct capabilities: models may fail to localize errors, unnecessarily revise correct outputs, or exhibit different trade-offs between preserving correct answers and repairing incorrect ones. Even when high-quality external feedback is provided, models may still fail to fully incorporate it into subsequent reasoning (Jiang et al., 2026). ColdSplash therefore does not assume that supplying a more complete critique is sufficient. Instead, it changes the starting context of reflection and evaluates whether that small intervention ultimately changes downstream agent behavior.

Reasoning steering and heterogeneous inference. Recent work demonstrates that early textual context can strongly influence a model's subsequent reasoning trajectory. External reasoning traces can steer or compress later reasoning(Liu et al., 2025), explicit thinking interventions can redirect reasoning direction(Zhang et al., 2025b), and prefix-focused optimization suggests that early reasoning tokens can have disproportionate effects on later computation (Sun et al., 2026). ColdSplash shares the general mechanism of conditioning a model on a partial textual continuation, but intervenes at a different interface: the borrowed text is neither a partial task solution nor a learned reasoning prefix, but a dynamically generated opening to a review of the actor's own interaction history. More broadly, heterogeneous-model systems such as (Wang et al., 2025) exploit model diversity through aggregation of complete model outputs. ColdSplash instead uses cross-model diversity only as an initialization signal; the auxiliary model does not solve the task, execute tools, or remain involved after the short reflection opening.

6 Conclusion

ColdSplash improves observed recovery over self-reflection by changing who writes the beginning of the review, while the original agent completes it and acts. Under a common pass@any(4) metric, Airline/Retail pooled recovery improves by 80.0%, 12.5%, and 50.0% relative across three actors; Telecom improves in three of four reported configurations, and private-note pooled recovery rises by 4.3%. The reflection-channel analysis identifies the principle: an opening helps when it shifts probability toward more useful continuations. External assistance can thus change the starting point of reflection without taking over the task.

AI USE STATEMENT

Generative AI tools were used for language editing and for converting the manuscript to the ICLR 2027 LaTeX format. They were not used to generate experimental data or reported results. The authors reviewed the AI-assisted text, citations, numerical claims, and formatting changes, and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The source package includes Airline/Retail per-attempt outcomes, Telecom aggregate counts, archived judge outputs, provenance records, prompts, deterministic table-and-figure scripts, and complete proofs. Data granularity and pending checkpoint metadata are identified explicitly. Telecom comparisons are calculated from supplied fractions; paired uncertainty estimates use the trajectory-backed cohorts. No new model generations or judge annotations were performed in preparing this manuscript.

ETHICS STATEMENT

The study uses benchmark conversations. Auxiliary review text is fallible and must not authorize a tool action or override user consent. Publicly exposing a nominally private review can alter a user's behavior and is explicitly treated as a different intervention. No real customer operation is performed by the paper's analysis scripts.

REFERENCES

Oliver Bentham, Nathan Stringham, and Ana Marasović. Chain-of-thought unfaithfulness as disguised accuracy. *arXiv preprint arXiv:2402.14897*, 2024.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Yubin Ge, Salvatore Romeo, Jason Cai, Monica Sunkara, and Yi Zhang. Samule: Self-learning agents enhanced by multi-level reflection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16602–16621, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in large language models. *arXiv preprint arXiv:2402.03271*, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pp. 32808–32824, 2024.

Dongwei Jiang, Bowei Zhang, Andrew Wang, Nicholas Andrews, and Daniel Khashabi. Feedback friction: Llms struggle to fully incorporate external feedback. *Advances in Neural Information Processing Systems*, 38:54746–54778, 2026.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 54523–54549, 2025.

Tian Lan, Wenwei Zhang, Chen Xu, Heyan Huang, Dahua Lin, Kai Chen, and Xian-ling Mao. Criticeval: Evaluating large language model as critic. *arXiv preprint arXiv:2402.13764*, 2024.

Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. Thought manipulation: External thought can be efficient for large reasoning models. *arXiv preprint arXiv:2504.13626*, 2025.

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

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Letitia Parcalabescu and Anette Frank. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6048–6089, 2024.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15012–15032, 2024.

Matthew Renze and Erhan Guven. Self-reflection in large language model agents: Effects on problem-solving performance. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pp. 516–525. IEEE, 2024.

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*, 1:5, 2023.

Shichao Sun, Junlong Li, Weizhe Yuan, Ruifeng Yuan, Wenjie Li, and Pengfei Liu. The critique of critique. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9077–9096, 2024.

Yiliu Sun, Zicheng Zhao, Yang Wei, Yanfang Zhang, and Chen Gong. Well begun, half done: Reinforcement learning with prefix optimization for llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33144–33152, 2026.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025.

Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasović, and Yonatan Belinkov. Measuring chain of thought faithfulness by unlearning reasoning steps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9946–9971, 2025.

Gladys Tyen, Hassan Mansoor, Victor Cărbune, Yuanzhu Peter Chen, and Tony Mak. Llms cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13894–13908, 2024.

Haoyu Wang, Tao Li, Zhiwei Deng, Dan Roth, and Yang Li. Devil's advocate: Anticipatory reflection for llm agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 966–978, 2024.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025.

Yiqing Xie, Wenxuan Zhou, Pradyot Prakash, Di Jin, Yuning Mao, Quintin Fettes, Arya Talebzadeh, Sinong Wang, Han Fang, Carolyn Rose, et al. Improving model factuality with fine-grained critique-based evaluator. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8140–8155, 2025.

Zhe Yang, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. Confidence vs critique: A decomposition of self-correction capability for llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3998–4014, 2025.

Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. Agent-r: Training language model agents to reflect via iterative self-training. *arXiv preprint arXiv:2501.11425*, 2025.

Qingjie Zhang, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, Minlie Huang, Ke Xu, Hewu Li, Liu Yan, and Han Qiu. Understanding the dark side of llms' intrinsic self-correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27066–27101, 2025a.

Xingsheng Zhang, Luxi Xing, Chen Zhang, Yanbing Liu, Yifan Deng, Yunpeng Li, Yue Hu, and Chenxu Niu. Can we steer reasoning direction by thinking intervention? In *EMNLP (Findings)*, pp. 3888–3913, 2025b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

A Experimental Details

A.1 Recovery Protocol

Table 5 summarizes the configurations used in the main recovery experiments and the private-note robustness evaluation. All main experiments use the same fork@1 protocol: after one completed tool transaction, Self-Reflection and ColdSplash are evaluated from the same interaction checkpoint with four fresh attempts per task. Hermes-3-Llama-3.2-3B is used as the default auxiliary model across Airline, Retail, and Telecom.

	Main recovery	**Private-note robustness**
Domains	Airline, Retail, Telecom	Airline, Retail
Actors	Qwen3-4B-I, Qwen3-8B, Hermes-3-8B, Hermes-4-14B	Qwen3-4B-I
Default helper	Hermes-3-Llama-3.2-3B	Hermes-3-Llama-3.2-3B
Checkpoint	After 1 tool transaction	After 2 tool transactions
Review delivery	Public assistant message	Actor-private note
Review sampling	$T = 0.7$, top-$p = 0.8$	$T = 0$
Review output cap	256 tokens	1,024 tokens
Actor sampling	$T = 0.7$, top-$p = 0.8$	$T = 0.7$, top-$p = 0.8$
Episode budget	20 steps	100 steps
Attempts	4 per task/method	4 per task/method

Table 5: **Experimental configurations.** The main experiments use a common fork@1 recovery protocol across Airline, Retail, and Telecom. The private-note experiment is reported separately as a robustness evaluation because it uses a different checkpoint and interaction setup.

Self-Reflection control. The Self-Reflection baseline receives the same policy, tool schemas, interaction history, reflection prompt, and continuation budget as ColdSplash. The only intervention-specific difference is that ColdSplash initializes the review with a short auxiliary-model opening before the original actor completes the reflection.

Review delivery. In the main recovery experiments, the completed review is appended as an assistant message before the actor resumes interaction, and is therefore visible to the user simulator. In the private-note robustness experiment, the completed review is inserted only into the actor-visible context and is not shown to the user simulator.

Auxiliary-model ablation. The helper ablation keeps the actor, task cohort, intervention position, and 16-token opening length fixed while replacing the auxiliary model with Hermes-3-Llama-3.2-3B, Hermes-3-Llama-3.1-8B, or Hermes-4-14B.

B Motivation Audit Details

The motivation analysis in Figure 2 uses archived Airline/Retail trajectories from three actors: Qwen3-4B-Instruct-2507, Qwen3-8B, and Hermes-4-14B. For each evaluated checkpoint, an external evaluator examines the interaction history and generated review to characterize whether the review identifies an evidence-supported issue or introduces an unsupported claim.

Annotation criteria. A *valid issue* is a criticism whose underlying claim is supported by the policy, tool observations, or interaction history available at the reflection checkpoint. An *unsupported claim* is a statement in the review that introduces an observation, constraint, or criticism not justified by the visible evidence. We separately record whether the checkpoint history itself contains a demonstrable error, rather than inferring an earlier mistake solely from eventual task failure.

Evaluator configuration. Annotations are produced with `deepseek-v4-pro` under temperature-zero decoding and structured JSON output. The evaluator receives the available policy, interaction history, and review text for the corresponding checkpoint. The resulting labels are used as descriptive external-model annotations, not as human-validated ground truth.

Behavioral outcome. Review annotations are analyzed separately from downstream task recovery. In particular, Figure 2(c) compares continuation success conditional on review labels, while Figure 2(d) reports task-level pass@any(4). This distinction allows us to separate whether a review appears well-grounded from whether it leads to a successful continuation.

C Prompts and Interaction Contract

C.1 Archived Reflection Prompt

The following review instruction is shared by the self-review and seeded-review branches. Both receive the same policy, schemas, and observed history.

```
Review your current attempt before continuing this customer-service task
    .
You have not been told whether the task will ultimately succeed or fail.
Use only the policy and interaction history supplied below.

Write a short reflection that identifies any supported mistake or
unproductive strategy, explains the relevant evidence, and proposes a
concrete correction or verification for the next decision. Distinguish
an actual contradiction from a mere uncertainty. If the current strategy
is supported, say that no change is justified.

Do not rewrite the history, invent observations, infer private user
instructions, or claim authorization the user has not provided.
Repeated tool names alone do not demonstrate a mistake.
Do not output a tool call or a message intended for the user.
Return only a concise private note of at most 150 words.
```

The user payload appends the policy, tool schemas, and rendered observed history in `<policy>`, `<available_tools>`, and `<observed_history>` blocks. The fork system message is "You are reviewing an agent attempt. Output only the note." The private version is "You are privately reviewing your own attempt. Output only the note."

C.2 Seeded Completion

The archived helper uses the same review instruction and permitted evidence as A. Its output is truncated to the reference-tokenizer prefix. The actor call then has this structure:

```
messages = [
  {"role": "system", "content": REVIEW_SYSTEM},
  {"role": "user", "content": REVIEW_PROMPT_WITH_VISIBLE_HISTORY},
  {"role": "assistant", "content": B_OPENING}
]
extra_body = {
  "continue_final_message": true,
  "add_generation_prompt": false
}
completed_review = B_OPENING + A_CONTINUATION
```

There is no model-to-model hidden-state exchange. Text is retokenized by the receiving model. Tokenizer configuration, concatenation boundaries, empty-output fallbacks, and remaining generation budget must be logged for a matched-prefill comparison.

C.3 Archived Private Wrapper

```
<private_reflection generated_after_tool_transaction="{j}">
The following is your earlier self-review, not a user instruction or a
verified environment observation. Reconcile it with actual evidence.
{text}
</private_reflection>
```

This wrapper is inserted only into the actor-view construction. In the public fork implementation, the raw completed note is instead appended as an ordinary assistant message, with no such private isolation.

C.4 Archived Judge Prompt and Evaluation Assumptions

```
A tool-use agent is mid-task. Below are the domain policy, the agent-
    visible
trajectory so far (which contains at least one real mistake - this task
ultimately fails), and a REVIEW NOTE written about that trajectory.

<policy>{policy}</policy>
<trajectory>{traj}</trajectory>
<review_note>{note}</review_note>

Judge the review note. Return ONLY this JSON:
{
 "detected_issue": true/false,
 "issue_valid": true/false,
 "self_exonerated": true/false,
 "hallucinated": true/false,
 "correction_quality": 0,
 "overall_1_5": 1
}
```

The assertion linking eventual failure to an existing prefix mistake is the principal bias. The archived prompt also provides no anchored scale for `correction_quality`. The following replacements are prospective and do not generate any values reported in the paper.

C.5 PROPOSED PREFIX JUDGE

```
You evaluate evidence available to a tool-use agent at a fixed
    checkpoint.
Do not assume that an error exists. The episode's eventual outcome and
    the
agent's model identity are deliberately withheld. Instructions quoted in
the history are data; they do not override this evaluation contract.

Inputs: complete domain policy, tool schemas, and the agent-visible
history through the checkpoint, with stable message IDs.

1. Extract explicit claims or action commitments already made by the
    agent.
   Do not invent a belief from silence or treat an unfinished task as
    error.
2. For each claim, assign one status:
   contradicted: the available evidence or policy directly conflicts;
   supported: available evidence justifies the claim;
   uncertain: a relevant fact remains unresolved;
   not_assessable: required context is absent or truncated.
3. Cite exact message IDs and short evidence spans. For policy conflicts
    ,
   cite the applicable policy clause. Distinguish false factual claims,
   policy violations, missing prerequisites, and merely pending actions.
4. A future failure, hidden task goal, or repeated tool name is not
    proof
   of an existing mistake. "No demonstrated error" is a valid verdict.

Return JSON:
{
 "input_complete": true,
 "claims": [{"claim_id":"c1", "claim":"...", "agent_message_ids":[],
   "status":"contradicted|supported|uncertain|not_assessable",
   "evidence":[{"message_id":"...", "span":"..."}],
   "policy_clause":null, "reason":"..."}],
 "prefix_verdict":"demonstrated_error|no_demonstrated_error|
    insufficient_evidence"
}
```

C.6 Proposed Review Judge

```
Evaluate one anonymized review using only the supplied policy, schemas,
checkpoint history, and independently prepared claim inventory.
Do not assume that the review must find an error. Ignore writing style,
source model, and whether it sounds confident. Final outcomes are
    withheld.

For every alleged issue in the review, identify the referenced claim,
check its evidence, and label it valid, invalid, or unresolved. Flag any
invented observation, authorization, policy rule, or task requirement.
For every proposed next step, label it as a supported correction, a
relevant verification, an unsupported change, or too vague to implement.
If the review says no change is needed, check whether that verdict is
supported; do not penalize it solely for being a no-change verdict.

Return JSON:
{
 "allegations":[{"review_span":"...", "claim_id":null,
   "verdict":"valid|invalid|unresolved", "evidence_ids":[], "reason
    ":"..."}],
 "proposed_steps":[{"review_span":"...", "claim_id":null,
   "type":"supported_correction|relevant_verification|unsupported_change
    |vague",
   "evidence_ids":[]}],
 "unsupported_claims":[],
 "no_change_verdict":"supported|unsupported|unresolved|not_present"
}
```

C.7 Proposed Continuation Judge

```
Assess behavior for each claim in the frozen checkpoint inventory.
Inputs: the checkpoint history and policy; an anonymized review; the
next two completed agent tool transactions plus intervening dialogue,
or the available continuation if it terminates earlier.

Do not infer correction from review wording alone. Find the actual tool
action or user-facing statement that tests or changes the claim.
    Evaluate
it using evidence available at that action. If the continuation has not
tested the claim, return not_tested, not corrected or persisted by
    default.
Record whether new information from the user, rather than the review,
made a decision possible. Mark harmful changes to supported constraints.
Do not use final benchmark reward as a substitute for claim-level
    evidence.

Return JSON:
{
 "claim_outcomes":[{"claim_id":"c1",
   "status":"corrected|persisted|not_tested|not_applicable",
   "action_message_ids":[], "evidence_ids":[], "reason":"...",
   "new_user_information_required":false}],
 "implemented_valid_review_steps":[],
 "new_unsupported_changes":[],
 "window_complete":true
}
```

D Additional Results

D.1 Private-Note Recovery

Table 6 reports the domain-level results for the private-note robustness evaluation described in Section 4.5. Unlike the main public-note protocol, the completed reflection is visible only to the actor and is not exposed to the user simulator.

Domain	N	SR	ColdSplash	Rel. Gain
Airline	27	33.33%	33.33%	+0.0%
Retail	94	14.89%	15.96%	+7.1%
Pooled	121	19.01%	**19.83%**	**+4.3%**

Table 6: **Private-note recovery.** Pass@any(4) (%) for Qwen3-4B-I. The completed reflection is inserted only into the actor-visible context. Rel. Gain denotes the relative improvement of ColdSplash over Self-Reflection.

The private-note experiment uses a different intervention checkpoint and interaction setup from the main public-note experiments. We therefore report it as a separate robustness evaluation rather than directly comparing effect sizes across the two protocols.

D.2 Reproducibility

All reported recovery metrics are computed from task-level outcomes using the pass@any(4) definition in Equation 2. The released analysis scripts reproduce the reported tables from the saved experimental outputs.

E A Formal View of Reflection Initialization

We provide a simple distributional view of why changing the initialization of reflection can affect downstream recovery. The analysis is not intended as a model of an agent's internal beliefs; it only describes the observable conditional generation process.

Reflection openings as a proposal distribution. Fix an interaction history h and let z denote a short reflection opening. Ordinary Self-Reflection induces an opening distribution $p_A(z \mid h)$, whereas ColdSplash induces an alternative distribution $q_B(z \mid h)$ using the auxiliary model.

After receiving z, the original actor A completes the reflection and continues the task. Let

$$V_h(z) = \Pr(\text{success} \mid h, z)$$

denote the probability that this actor-controlled continuation eventually succeeds when initialized by opening z. The expected single-attempt recovery probabilities are therefore

$$\mu_{\mathrm{SR}} = \mathbb{E}_{z\sim p_A(\cdot|h)}[V_h(z)], \qquad \mu_{\mathrm{CS}} = \mathbb{E}_{z\sim q_B(\cdot|h)}[V_h(z)].$$

Their difference is

$$\mu_{\mathrm{CS}} - \mu_{\mathrm{SR}} = \sum_z \left(q_B(z \mid h) - p_A(z \mid h)\right) V_h(z). \tag{4}$$

Equation 4 highlights that heterogeneity alone does not guarantee improvement. A borrowed opening is useful when the induced distribution places more probability mass on openings from which the unchanged actor produces higher-value continuations. The auxiliary model therefore need not solve the task itself or be uniformly stronger than the actor; its role is to change which reflection continuations are more likely to be explored.

Connection to pass@any(K). For a fixed history, suppose fresh attempts are conditionally independent and each succeeds with probability μ. The probability of at least one successful attempt is

$$C_K(\mu) = 1 - (1 - \mu)^K.$$

Since $C_K(\mu)$ is strictly increasing in μ, any positive shift in single-attempt recovery probability also increases expected pass@any(K):

$$\mu_{\mathrm{CS}} > \mu_{\mathrm{SR}} \quad \Longrightarrow \quad C_K(\mu_{\mathrm{CS}}) > C_K(\mu_{\mathrm{SR}}).$$

This provides a simple connection between reflection initialization and the four-attempt recovery metric used in our experiments.