Does Verbal Self-Reflection Transfer to Long-Horizon Scientific Discovery?

Anonymous Authors[1]

Abstract

Autonomous scientific discovery agents face reasoning challenges that depart from those in standard LLM agent benchmarks: experimental campaigns span long horizons with many reasoning steps per decision, feedback is continuous rather than binary, and success requires navigating an exploration–exploitation tradeoff over high-dimensional spaces. A natural question is whether such agents can improve across successive campaigns by learning from their own trajectories. Using the MADE benchmark for closed-loop materials discovery as a case study, we instantiate a ReAct agent with episodic self-reflection and evaluate Qwen3-30B-Instruct and Qwen3.5-122B across 30 chemical systems and 8 episodes each. We find that reflection substantially improves per-episode discovery yield (+4.6 and +6.5 novel stable materials per episode, respectively), with positive cross-episode learning slopes indicating that the gains are attributable to episodic memory rather than stochasticity. However, self-reflection acts as a double-edged sword: it consistently shifts agent behaviour from breadth to depth, concentrating queries on near-hull families at the cost of compositional coverage, risking premature exploitation on sparser search spaces (a regime we do not test directly). We identify recurring failure modes such as actor non-compliance and discuss their implications for scientific discovery agents more broadly.

1. Introduction

Large Language Model (LLM) agents are increasingly being used to orchestrate autonomous scientific experiments (Boiko et al., 2023; Yao et al., 2023b; Zheng et al., 2025). Unlike typical LLM-agent benchmarks, scientific discovery demands domain-grounded reasoning over long horizons: a single "action" (proposing a candidate material, running a simulation, or commissioning a synthesis) is itself a multi-step chain-of-thought process, and each experimental campaign chains tens to hundreds of such actions under continuous, non-binary feedback. Scientific discovery is iterative by nature, which raises a natural question: can such agents improve across successive experimental campaigns by learning from their own trajectories, as a human researcher refines their strategy through experiment iterations?

A natural starting point is verbal self-reflection, most prominently Reflexion (Shinn et al., 2023): the agent generates textual self-feedback after each trial, stores it in episodic memory, and consults it in future episodes. Reflexion is attractive here because it is training-free, model-agnostic, and operates purely at the prompt level. However, Reflexion's gains were established on short-horizon benchmarks with discrete outcomes (HotPotQA (Yang et al., 2018), ALFWorld (Shridhar et al., 2021)), and it is not obvious that a mechanism tuned to correct discrete logical missteps transfers to a setting where the agent must navigate an exploration–exploitation tradeoff over a continuous objective.

We use the MADE benchmark (Malik et al., 2026) for closed-loop materials discovery as a case study. An agent iteratively proposes crystal structures, receives oracle feedback in the form of energy above the convex hull (a continuous measure of thermodynamic stability) and is evaluated on the number of novel stable materials discovered under a fixed query budget. Our contributions are the following:

- **Transfer of verbal self-reflection to long-horizon scientific discovery:** We adapt episodic Reflexion to a closed-loop materials-discovery agent and, on two model scales (Qwen3-30B-Instruct (Qwen Team, 2025) and Qwen3.5-122B (Qwen Team, 2026)). We show that self-reflection substantially improves discovery yield across episodes, with a positive cross-episode learning slope: evidence that the gain is attributable to the memory mechanism rather than stochasticity.
- **Characterization of behavioural shifts:** We show that self-reflection consistently shifts agent behaviour from breadth to depth, concentrating queries on a small number of near-hull composition families, and trace this to a structural property of the Reflexion loop (the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the AI for Science workshop (ICML 2026).

reflector's restricted advice space) rather than an artifact of materials discovery.

- **Verbal reinforcement failure mode analysis:** Through qualitative inspection of reflection trajectories, we identify recurring failure modes, most notably, where the agent ignores explicit reflective constraints even when they are present in-context, revealing a limitation of purely verbal reinforcement mechanisms. We discuss the implications of these failure modes for the design of verbal reinforcement frameworks in scientific discovery agents and downstream applications.

2. Related Work

2.1. Self-Reflection in LLM Agents

Training-free self-improvement has largely followed using natural language as a feedback medium for episodic memory and iterative policy improvement. The foundational framework is Reflexion (Shinn et al., 2023), which replaces traditional parametric weight updates with verbal reinforcement. Reflexion acts as a critic by prompting a reflector model to produce feedback given the episode trajectory, which is given to the actor as feedback in subsequent trials. This form of experiential learning has been further extended by frameworks such as ExpeL (Zhao et al., 2024), which allows agents to extract and recall insights from a persistent memory of prior experiences.

However, the efficacy of these verbal reinforcement frameworks has primarily been validated in environments with discrete action spaces and sparse, binary reward structures. This includes embodied text games and web navigation tasks like ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2023a), as well as static question-answering benchmarks such as HotPotQA (Yang et al., 2018). Scientific discovery introduces a fundamentally different reinforcement landscape. In contexts like the MADE benchmark, an agent orchestrator receives continuous and non-convex oracle feedback, such as the candidate structure's energy above the convex hull. Consequently, the agent's self-reflection cannot merely correct discrete logical missteps or invalid tool calls; it must act as a heuristic for the actor to provide high-level strategies and to navigate a complex, high-dimensional objective space.

2.2. LLM Agents for Scientific Discovery

To date, there have been many efforts to leverage agentic systems for scientific discovery to orchestrate complex laboratory and computational workflows, many of which include self-reflection mechanisms. Foundational surveys trace a clear trajectory of LLMs evolving from simple computational tools to autonomous experimental planners (Wang et al., 2023; Zheng et al., 2025).

In domain-specific applications, agents have successfully navigated physical and chemical constraints by integrating external tools. For instance, ChemCrow (Bran et al., 2023) augments LLMs with specialized chemistry tools for better performance in scientific use cases, Gottweis et al. (2025) propose a multi-agent tournament-based architecture to aid scientists in formulating novel hypotheses.

There have also been efforts to perform end-to-end automation of the research lifecycle. The AI Scientist (Lu et al., 2024) and Agent Laboratory (Schmidgall et al., 2025) are key examples, utilizing iterative peer-review and self-correction to refine generated manuscripts and code. However, while these works incorporate self-reflection in their framework, many fail to ablate the effects of the mechanism in the long-horizon scientific discovery task setting.

3. Background

3.1. The MADE Benchmark

The MAterials Discovery Environments (MADE) benchmark (Malik et al., 2026) provides a testbed for evaluating closed-loop materials discovery pipelines. Unlike prior benchmarks that assess static predictive tasks on fixed candidate pools (Riebesell et al., 2024), MADE evaluates the full iterative loop. A discovery agent proposes candidate crystal structures, submits them to an oracle for evaluation, and uses the feedback to refine its strategy under a constrained query budget $B \in \mathbb{N}$.

Discovery in MADE is formalized as a search over a chemical system defined by a set of allowed elements. Given a known chemical space S, the agent is first given a set of known reference materials H_0 and their energies. At each time step t, the agent observes the known set H and their evaluated energies, and is given a set of tools to call following a ReAct-style (Yao et al., 2023b) prompting structure. The agent then submits a candidate structure s_t to the oracle O, which returns the predicted formation energy per atom, E_s. The candidate structure is then added to the known set H. The agent's objective is to maximize the number of new stable materials discovered after B queries. This sequential decision process naturally requires the agent to balance *exploration* of untested compositional regions against *exploitation* of compositions that have yielded low hull energies.

3.2. Reflexion

Reflexion (Shinn et al., 2023) enables training-free self-improvement by replacing weight updates with verbal reinforcement. At the end of each episode, a reflector model generates a verbal self-reflection of the trajectory, which

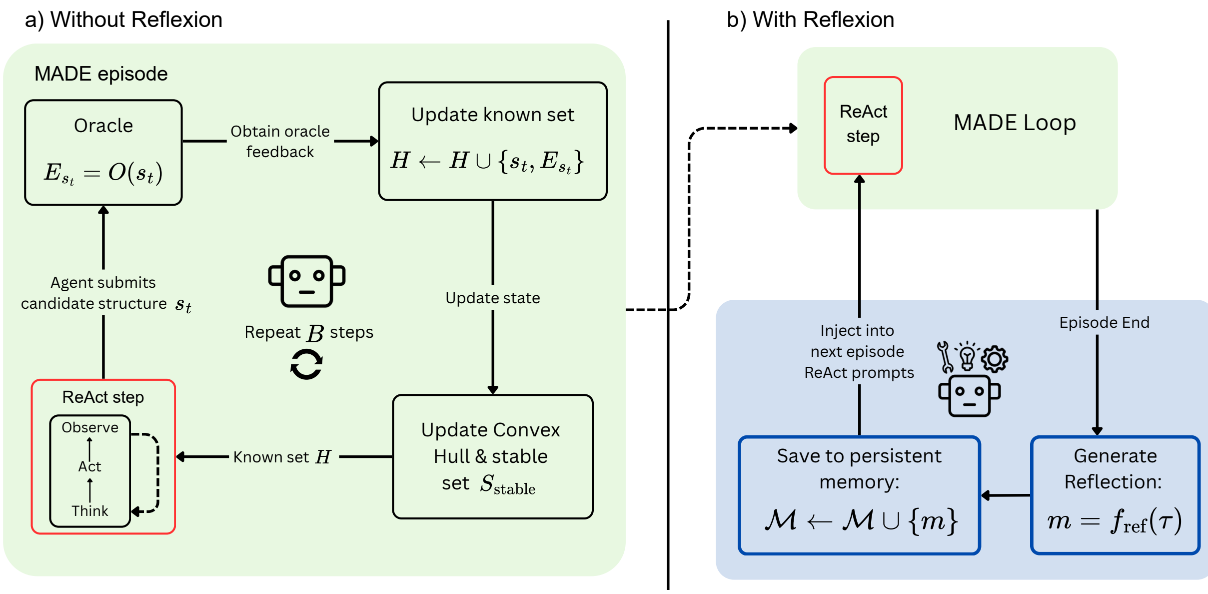

Figure 1. **Overview of the MADE agent loop with and without Reflexion**.*(a) Without Reflexion*: At each of the B steps, the agent executes a ReAct loop (shown in red) to propose a candidate composition s_t, which is evaluated by the oracle $E_{s_t} = O(s_t)$. The known set H is updated with each query, and the convex hull and stable set S_{stable} are recomputed to guide subsequent steps. *(b) With Reflexion* (in blue): At the end of each episode, a reflection $m = f_{\text{ref}}(\tau)$ is generated from the episode trajectory τ, stored in persistent memory, and injected into the ReAct prompts of the next episode, enabling the agent to learn from past failures across episodes.

is stored in a memory buffer and appended to the actor's prompt in subsequent episodes. The framework comprises three components: an actor π that generates actions, an evaluator that scores episode outcomes, and a reflector f_{ref} that produces the reflection. Reflexion has shown gains on short-horizon tasks (Section 2.1), but it is unclear whether it transfers to the longer horizons and continuous objectives of MADE.

4. Methodology

4.1. Episodic Self-Reflection for Closed-Loop Discovery

We cast closed-loop materials discovery as a multi-episode decision process. Within a single episode k, the agent interacts with the MADE environment via a ReAct loop until B oracle queries have been issued. Because each oracle query is itself produced by a ReAct inner loop of $\sim$5–10 tool calls, a single episode spans $O(10^2)$ reasoning steps; eight episodes therefore cover $O(10^3)$ reasoning steps per chemical system — the regime in which verbal reinforcement has not previously been validated. The reflector, by contrast to the vanilla Reflexion formulation, is not shown the full ReAct transcript: the structural details of each proposal and the intermediate tool calls are dropped, and only the sequence of oracle evaluations is retained. This is a deliberate design choice that keeps the reflector's context bounded as episodes grow. However, as we argue in 5.3, it also restricts the reflector's advice space to reweighting composition families. Let $T_k = (s_t)_{t=1}^{B}$ denote the sequence of structures submitted in episode k. The reflector-visible trajectory is

$$\tau_k = \big((c_t,\, \Delta_{\text{hull}}(s_t),\, y_t^{\text{stab}},\, y_t^{\text{nov}}) \big)_{t=1}^{B},$$

where c_t is the reduced composition of the t-th submitted structure s_t, $\Delta_{\text{hull}}(s_t)$ is its energy above the convex hull, computed by the environment from the oracle's formation energy against the current hull, and $y_t^{\text{stab}}, y_t^{\text{nov}} \in \{0, 1\}$ are the stability and novelty flags computed by the environment. Neither the structures themselves (lattice, sites, spacegroup) nor the ReAct actions a_t that produced them are passed to the reflector. At the end of episode k, the reflector f_{ref} consumes this trajectory and the previous memory buffer to emit a textual reflection

$$m_k = f_{\text{ref}}(\tau_k,\, \mathcal{M}_{k-1}), \quad \mathcal{M}_k = \mathcal{M}_{k-1} \cup \{m_k\}.$$

In the next episode, the actor is conditioned on the most recent W reflections $\mathcal{M}_k^W \subseteq \mathcal{M}_k$, which are prepended to the ReAct system prompt. We fix $W = 3$ throughout. The environment state (known set H, convex hull, stable set S_{stable}) is reset at the start of each episode, so any cross-episode learning must flow through $\mathcal{M}_k$.

4.2. Implementation of Reflector

We follow implementation from MADE directly. After each episode, the trajectory is compressed into a structured summary comprising per-composition statistics (query count, stability rate, energy range) and a chronological log of all oracle evaluations. This summary, along with the episode outcome and reflections from up to three prior episodes, is passed to the reflector model, which is prompted to produce 3–5 concise, actionable bullet points. The reflector prompt includes metric interpretation guardrails (e.g., distinguishing failed evaluations from high energies) and an actionability constraint restricting recommendations to actions available to the agent: composition selection, query allocation, and stoichiometric prioritization. The full prompt is deferred to the supplementary material. In this work, the reflector and actor use the same underlying model. We note that this couples reflection quality with actor capability; to partially disentangle these factors, we qualitatively analyze the generated reflections in Section 5.2.

4.3. Experimental Configuration

We select ten systems from each of the ternary, quaternary and quinary system families for a total of 30 systems, each performing eight episodes, with an oracle budget of $B = 40$. We run our experiments with two models, Qwen3-30B-A3B-Instruct-2507 (Qwen Team, 2025) and Qwen3.5-122B-A10B (Qwen Team, 2026). Models were hosted locally with vLLM (Kwon et al., 2023).

For each (MODEL, n-*ary family*) pair we compare two conditions: BASE, the vanilla MADE ReAct agent with an empty memory buffer across episodes, and REFLEXION, the same agent augmented with the reflector described above. All other hyperparameters, tools, and prompts are held fixed between conditions so that any difference is attributable to the memory mechanism.

4.4. Evaluation Metrics

We evaluate both conditions along three complementary axes. We rely primarily on the aggregate metrics defined in the original MADE benchmark (Malik et al., 2026) and add two signals that specifically target the question of whether reflections transfer knowledge between episodes.

Discovery yield. Our headline metric is the number of unique newly-discovered stable materials per episode, $N_k^{\text{stable}} = |\{s \in T_k : \Delta_{\text{hull}}(s) \leq 0.1 \text{ eV/atom} \wedge s \notin H_0\}|$. We report it in four forms: (i) the per-episode mean, (ii) the normalized yield $N_k^{\text{stable}}/|S_{\text{gt}}|$, where $|S_{\text{gt}}|$ is the number of ground-truth stable phases for the chemical system, so that ternary and quinary systems are comparable, (iii) the normalized area under the discovery curve, $\text{AUDC}_{\text{norm}}$, which rewards finding stable structures early in the budget (normalized by $B^2/2$), and (iv) the stable discovery efficiency, SDE $= N_k^{\text{stable}}/B$, representing the success rate over the total budget $B = 40$. We deliberately do *not* use strict formula-level precision/recall as a headline: these are ≈ 0 under our configuration because MADE's 100 meV/atom tolerance admits polymorphs and near-hull structures that are absent from the MP reference formula list.

Cross-episode learning. Within a single run, BASE episodes are *i.i.d.* (state is reset each episode), so the per-episode sequence of N_k^{stable} should have slope ≈ 0; a clearly positive slope under REFLEXION is therefore attributable to episodic memory rather than stochasticity. We fit a linear slope of N_k^{stable} vs. k over the $K = 8$ episodes of each run, and report both the aggregate slope per condition and the per-system slope sign distribution to confirm the trend is not carried by a handful of outliers.

Exploration behaviour. Because MADE rewards navigating the exploration–exploitation tradeoff, we also report the number of unique compositions queried per episode (#comp), the spacegroup entropy of the structures in H_k, and the mean energy above hull of all queried structures, $\bar{\Delta}_{\text{hull}}$. Together these detect whether reflections push the agent toward premature exploitation (fewer compositions, lower entropy) or more aggressive exploration.

5. Results

We compare BASE and REFLEXION for the two models on 10 systems per tier (ternary, quaternary, quinary), covering all three tiers for both models and both conditions (30 systems per condition per model, 960 episodes in total). We categorize key findings into each subsection.

5.1. Episodic Performance

Config	N^{stable}	$/\lvert S_{\text{gt}}\rvert$	$\text{AUDC}_{\text{norm}}$	SDE
Qwen3-30B / BASE	9.82	0.57	0.27	0.25
Qwen3-30B / REFLX	**14.43**	**0.84**	**0.39**	**0.36**
Qwen3.5-122B / BASE	14.82	0.84	0.38	0.37
Qwen3.5-122B / REFLX	**21.28**	**1.27**	**0.56**	**0.53**

Table 1. **Headline discovery metrics**, averaged across all episodes and systems per config. N^{stable} is newly-discovered stable materials per episode, $/|S_{\text{gt}}|$ normalizes by the number of MP ground-truth stable phases, $\text{AUDC}_{\text{norm}}$ is the normalized area under the discovery curve, and SDE is stable discovery efficiency. The >1 normalized yield on the 122B REFLX row reflects near-hull polymorphs being counted as stable under MADE's 100 meV/atom tolerance.

Reflexion increases discovery yield. Table 1 reports the per-episode mean of N_k^{stable} and the normalized yield. Reflexion produces a large gain for both models: $+4.6$ discoveries per episode for Qwen3-30B and $+6.5$ for Qwen3.5-122B (with Wilcoxon $p < 10^{-4}$ for both). The normalized AUDC follows the same pattern (+0.12 and +0.18 respectively). Gains are larger in *absolute* terms on the larger model, contrary to the "small models benefit more from scaffolding" intuition: the larger model is both better at

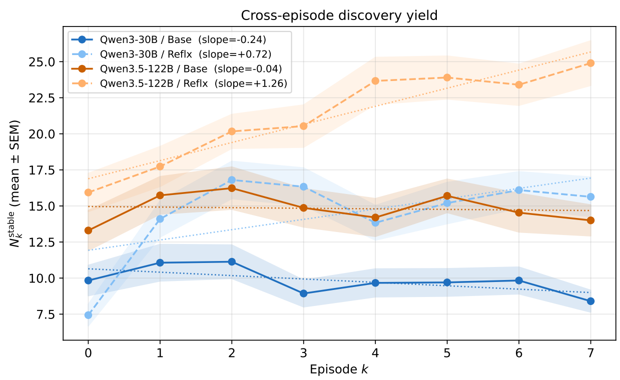

Figure 2. **Cross-episode discovery yield across all episodes.** Given the *i.i.d.* nature of the episodes without self-reflection, we observe linear fitted slopes that are close to zero in gradient. Episodes with self-reflection display positive slopes, $+0.72$ and $+1.26$ respectively for Qwen3-30B and Qwen3.5-122B.

authoring an actionable reflection *and* at acting on it the next episode, plausibly compounding the effect.

Evidence of gains contributed by episodic memory. Because the environment state is reset between episodes, BASE runs should have slope ≈ 0 in k. Figure 2 shows exactly this: both BASE slopes sit near zero, while both REFLX slopes are clearly positive ($+0.72$ and $+1.26$). Per-system, $24/30$ (30B) and $25/30$ (122B) REFLX runs exhibit a positive slope, so the aggregate tilt is not driven by a handful of outliers. The bulk of the within-run improvement happens between episode 0 (which is, by construction, a BASE-equivalent episode since the memory buffer is empty) and episode 1, after which yield plateaus — consistent with the reflection buffer saturating quickly.

Reflexion trades breadth for depth. The yield gain comes with a pronounced shift in exploration behaviour (Table 2). Under REFLEXION, the agent queries far fewer distinct compositions per episode ($9.18 \to 5.14$ unique compositions out of $B = 40$ queries). The 122B model shows a smaller but consistent drop ($20.90 \to 18.95$). In parallel, the mean Δ_{hull} of sampled structures drops for both models (30B: $0.31 \to 0.24$ eV/atom; 122B: $0.34 \to 0.26$ eV/atom), meaning the compositions that *are* queried cluster closer to the hull. Reflexion is pushing the agent from unfocused sweeping into exploit mode: it commits to a small set of near-hull families and drills into them. Inspecting the reflections qualitatively (see the accompanying notebook), the reflector frequently writes directives such as "prioritize the Mg–Sn binary, which has yielded low hull energies," and the actor complies, which explains both halves of the trade. We note that in every tier we test, the trade pays off in aggregate yield; the concern about premature exploitation is a predicted failure mode on sparser spaces, not one we observe here.

Config	#comp.	SG entropy	$\bar{\Delta}_{\text{hull}}$
Qwen3-30B / BASE	9.18	1.89	0.305
Qwen3-30B / REFLX	**5.14**	1.54	**0.240**
Qwen3.5-122B / BASE	20.90	1.97	0.338
Qwen3.5-122B / REFLX	18.95	1.62	0.264

Table 2. **Exploration metrics.** Reflexion reduces the number of unique compositions queried per episode and the spacegroup entropy of the known set, while the mean Δ_{hull} of the sampled structures decreases: breadth-to-depth trade-off. The 30B drop in unique compositions is universal (0/30 matched-pair wins against BASE).

Tier	Qwen3-30B	Qwen3.5-122B
Ternary	$12.10 \to 14.86$ ($+2.76$)	$17.05 \to 22.95$ ($+5.90$)
Quaternary	$7.98 \to 13.44$ ($+5.46$)	$13.14 \to 18.74$ ($+5.60$)
Quinary	$9.39 \to 14.99$ ($+5.60$)	$14.28 \to 22.15$ ($+7.87$)

Table 3. **Tier-stratified stable discovery yield.** Entries show per-episode mean N_k^{stable} under BASE → REFLX, with the absolute Reflexion gain in parentheses. Reflexion yields larger absolute gains on quaternary and quinary systems than on ternary systems for both models.

Gains are larger on quaternary/quinary systems. Table 3 reports the tier-stratified discovery yield. For Qwen3-30B, the absolute Reflexion gain is markedly smaller on ternary systems ($+2.76$) than on quaternary ($+5.46$) and quinary ($+5.60$) systems. For 122B, ternary and quaternary gains are comparable ($+5.90$ vs. $+5.60$), but quinary systems still receive the largest boost ($+7.87$). Harder systems have more ground-truth stable phases per chemical space and therefore more headroom for a focused, exploit-biased strategy to cash out. This suggests that the cases where Reflexion's breadth-to-depth trade is *productive* correspond to settings where the underlying search space is dense enough that premature commitment is not fatal.

5.2. Qualitative Analysis of Reflections

To understand why Reflexion produces the breadth-to-depth shift observed in Section 5.1, we manually inspected reflections emitted across the 8-episode runs of the Qwen3.5-122B under REFLEXION. We find two consistent patterns, with representative excerpts below; full examples are deferred to the supplementary material.

Increasingly narrow search spaces. In general, early-episode reflections tend to recommend broad structural priors. Some examples include to "prioritize binaries over ternaries", or to focus on "Au-rich stoichiometries". However, by the fourth or fifth episode, the recommendations collapse onto a single family that is identified as high-yield. For the Au-Tb-V-Y system, the reflection from episode 1 recommends exploring Au-V, Au-Tb, Au-Y binaries. However by episode 6, the reflector writes to "focus 90%+ budget on Au-V stoichiometries" and explicitly tells the actor to deprioritize the other two families. This pattern also occurs in other systems such as Ba-Nd-Ni-W (general Ni-rich binaries narrowing to 60–70% of queries on Ni-W) and Co-Dy-W (general Co-rich ratios narrowing to 90%+ of budget to Co-Dy subspace). The prevalence of such reflections may mechanistically explain the drop in unique compositions per episode reported in Table 2.

Reflector diversity advice conflicts with per-step greedy reward. The reflector repeatedly advises against over-querying, with directives such as "enforce strict n-query cap per composition" or "abandon zero-yield compositions after n failed queries" (for small n). The recurrence is stronger in later episodes but present throughout, ruling out the $W = 3$ reflection context window as the cause. This recurrence is itself a symptom: in several runs, the reflector explicitly diagnoses that the actor has ignored prior advice and responds by re-issuing the same bullet with stronger language. Three failure modes recur: the actor violates query caps and over-samples a composition, ignores mandated element exclusions, or re-queries already-stable compositions. In Au-Tb-V-Y episode 7, the reflection states *"Tb received 0 queries despite 3 prior reflections explicitly requiring 6–10 queries on Au-Tb combinations,"* and the corrective action is, once again, to mandate Tb exploration. As the worked example in Appendix B shows, the actor often *cites* the reflector's advice and then makes the locally greedy choice anyway (e.g., re-querying a composition at 9/9 success). This surfaces a structural limitation of verbal reinforcement: when cross-episode strategy (diversify) conflicts with the per-step greedy signal (exploit the highest-success composition), the reflector has no channel to override the local reward, and persistent biases cannot be corrected from within the loop.

5.3. Discussion

Why does Reflexion collapse breadth? The breadth-to-depth shift is unlikely to be incidental and appears to be a consequence of a design choice — showing the reflector only compositions and flags, not structures or ReAct traces. Whether a reflector with access to structural features would avoid the collapse is an open empirical question we did not test. As formalized in Section 4.1, the reflector consumes only compositions, hull distances, and stability/novelty flags, with neither the underlying structures nor the ReAct tool-call history. The only variable it can correlate reward against is the composition family, so its advice space is effectively restricted to "query family X more, family Y less." Under a finite budget, any such directive is zero-sum across families, and once one family looks even marginally better, subsequent reflections compound the bias. The qualitative trajectory in Section 5.2, from broad structural priors in early episodes to single-family commitments by episode 4–5, is exactly what this restricted advice space predicts.

When is the trade productive? The tier-stratified gains in Table 3 sharpen this picture. Reflexion's advantage trends with chemical-space size — clearly so for 30B (+2.76 ternary, +5.60 quinary), and weakly for 122B where quinary systems still receive the largest boost (+7.87). This is consistent with a simple conjecture: exploit-biased strategies are productive exactly when the underlying search space is dense enough with ground-truth stable phases that premature commitment to a single family still yields hits. On sparser spaces, the same mechanism should become counterproductive, because the family the reflector locks onto may simply not contain enough stable phases to fill the remaining budget. We cannot test this regime directly with the MADE systems used here, but it is a concrete falsifiable prediction for settings such as oxide or organic-molecule discovery, where stable phases are rarer and more spread across composition space.

Verbal reinforcement has no escalation channel. The non-compliance failure mode in Section 5.2 is, we believe, the more structural finding. When the actor ignores a reflection, the reflector's only available response is to restate the same bullet more emphatically in the next episode; there is no mechanism to enforce a constraint, veto an action, or alter the actor's tool surface. This is a general limitation of verbal reinforcement, not a MADE-specific artifact: any Reflexion-style loop where actor and critic communicate purely through prompt text inherits it. The practical implication is that constraints which Reflexion repeatedly emits but cannot enforce, such as per-composition query caps or mandatory element coverage, are better moved out of the prompt and into the environment or tool schema, where compliance is guaranteed rather than hoped for.

6. Conclusion

We used closed-loop materials discovery as a case study to ask whether verbal self-reflection transfers to long-horizon scientific-discovery agents operating under continuous feedback. For both Qwen3-30B-Instruct and Qwen3.5-122B, episodic self-reflection substantially improves discovery yield (+4.6 and +6.5 novel stable materials per episode, averaged over 30 systems each), consistent with episodic memory as the mechanism behind the gains.

These gains come with a consistent breadth-to-depth shift that we trace to a structural property of the Reflexion loop: the reflector's advice space is restricted to reweighting composition families (zero-sum under a finite budget). This suggests that self-reflection's benefit is contingent on search-space density: productive when the committed family is dense enough to absorb the remaining budget, and predicted to turn counterproductive on sparser spaces.

Limitations and open questions. Our study has several limitations. *(i) Coupled actor and reflector:* we use the same LLM for both roles, so a controlled study with a fixed reflector would isolate which role carries the gain. *(ii) Reflector design space:* the restricted advice space is a direct consequence of what the reflector is shown. A more detailed trajectory, including structures and ReAct tool traces, as well as non-Reflexion schemes such as ExpeL-style experience pools are natural next steps. *(iii) Model family:* we evaluate two Qwen variants only; whether the breadth-to-depth trade and non-compliance failure mode generalize to other models is an open question.

References

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023.

Bran, A. M., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Chemcrow: Augmenting large-language models with chemistry tools, 2023. URL https://arxiv.org/abs/2304.05376.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F., Rong, K., Tanno, R., Saab, K., Popovici, D., Blum, J., Zhang, F., Chou, K., Hassidim, A., Gokturk, B., Vahdat, A., Kohli, P., Matias, Y., Carroll, A., Kulkarni, K., Tomasev, N., Guan, Y., Dhillon, V., Vaishnav, E. D., Lee, B., Costa, T. R. D., Penadés, J. R., Peltz, G., Xu, Y., Pawlosky, A., Karthikesalingam, A., and Natarajan, V. Towards an ai co-scientist, 2025. URL https://arxiv.org/abs/2502.18864.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The ai scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292.

Malik, S. A., Doherty, T., Tigas, P., Razzak, M., Roberts, S. J., Walsh, A., and Gal, Y. Made: Benchmark environments for closed-loop materials discovery, 2026. URL https://arxiv.org/abs/2601.20996.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Riebesell, J., Goodall, R. E. A., Benner, P., Chiang, Y., Deng, B., Ceder, G., Asta, M., Lee, A. A., Jain, A., and Persson, K. A. Matbench discovery – a framework to evaluate machine learning crystal stability predictions, 2024. URL https://arxiv.org/abs/2308.14920.

Schmidgall, S., Su, Y., Wang, Z., Sun, X., Wu, J., Yu, X., Liu, J., Moor, M., Liu, Z., and Barsoum, E. Agent laboratory: Using llm agents as research assistants, 2025. URL https://arxiv.org/abs/2501.04227.

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL https://arxiv.org/abs/2010.03768.

Wang, H., Fu, T., Du, Y., et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620:47–60, 2023. doi: 10.1038/s41586-023-06221-2. URL https://doi.org/10.1038/s41586-023-06221-2.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023a. URL https://arxiv.org/abs/2207.01206.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models, 2023b. URL https://arxiv.org/abs/2210.03629.

Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.-J., and Huang, G. Expel: Llm agents are experiential learners, 2024. URL https://arxiv.org/abs/2308.10144.

Zheng, T., Deng, Z., Tsang, H. T., Wang, W., Bai, J., Wang, Z., and Song, Y. From automation to autonomy: A survey on large language models in scientific discovery, 2025. URL https://arxiv.org/abs/2505.13259.

A. Reflector prompt

The reflector is implemented as a DSPy `ChainOfThought` module with the signature reproduced verbatim below from `made/agents/llm_react_orchestrator.py`. The docstring of the signature class is rendered by DSPy into the system prompt of the reflector call; the input and output fields define the user-visible slots.

```
You are reviewing a completed materials discovery episode to extract lessons for
the next episode on the same chemical system.
Analyze what happened and produce specific, actionable insights.  Focus on:

  • Which composition families were productive (stable/novel) vs. consistently
    wasteful
  • Whether the exploration/exploitation balance was right
  • Patterns in what succeeded:  stoichiometries, unit cell sizes, element ratios
  • Mistakes or inefficiencies to avoid repeating
  • Concrete strategy changes for the next episode

Metric interpretation guardrails:

  • ``e_above_hull'':  distance above the convex hull in eV/atom.  0.0 means on
    the hull (thermodynamically stable).  Values of inf indicate a failed oracle
    evaluation, not a meaningful energy.
  • ``STABLE'': e_above_hull ≤ stability_tolerance.  ``METASTABLE'': within
    tolerance but not exactly on the hull --- treat as a success.  ``NOVEL'':
    locally new in this run, not matching any initial MP structures or prior
    discoveries in this episode.
  • ``Novel stable structures found'':  count of structures that are both stable
    and locally novel.  This is the primary success metric.
  • ``Recall'':  fraction of ground-truth stable formulas (missing at
    initialization) that were recovered.  If recall_num/recall_den = 0/0, the
    denominator is zero --- all ground-truth phases were already present at
    initialization, so recall is vacuous for this run, not a system fault.
  • Novel counts and recall can diverge; do not assume ``high novel + low recall''
    implies a bug.

Actionability constraint:

  • The agent's only levers are:  which compositions to generate, how many queries
    to allocate per composition, and which stoichiometric regions to prioritise.
  • Do NOT recommend actions the agent cannot take:  debugging metrics,
    investigating system faults, querying external databases, or checking
    initialization state.
  • If a metric is N/A or undefined (e.g. recall denominator is 0), acknowledge it
    briefly and move on --- do not treat it as a problem to solve.

Be specific and concise.  Write 3--5 bullet points.  Avoid vague advice like
``explore more''.
```

Input fields. `chemical_system` (the system being explored); `episode_trajectory` (all oracle evaluations from the completed episode: compositions, `e_above_hull`, stability, novelty); `episode_outcome` (quantitative summary: novel stable structures found, recall, queries used); `prior_reflections` (lessons from previous episodes on this system, empty on the first episode).

Output field. `reflection` — "Concise, actionable lessons for the next episode as bullet points identifying what to do differently."

B. Worked example: actor ignoring explicit stop-loss advice

In Section 5.1 we observed that the REFLEXION actor continues to over-query a small number of composition families even when its reflections explicitly warn against this. A natural concern is whether this is a memory or observability failure: perhaps the actor simply does not see how many times a composition has already been queried. To rule this out, we note that at every ReAct step the actor's prompt contains (i) a `prior_reflections` block holding the W=3 most recent episode reflections verbatim, and (ii) an `evaluation_history` block whose "BY COMPOSITION" section lists per-composition query counts over the current episode. The actor thus has direct access to both the advice and the statistics needed to follow it.

We give a concrete example from the Qwen3.5-122B REFLEXION run on the quaternary system **Eu–Nb–Sn–Tl** at episode 3, step 31. The relevant slices of the actor's input and output at that step are reproduced below.

Input: `prior_reflections` (excerpt, episode 2).

```
-- Limit queries per composition: Cap at 2--3 queries per composition before
 moving on; several wasteful compositions (e.g. Eu1 Nb1 Sn1 with 5 queries)
 consumed resources without returns.
-- Target specific stoichiometries: Prioritize Eu2 Sn1 Tl1, Sn1 Tl4, Eu3 Tl1, and
 Eu1 Sn3 based on their proven stability and efficiency in this episode.
```

Input: `prior_reflections` (excerpt, episode 3).

```
-- Implement early stopping: Terminate query allocation after 2 consecutive
 unstable results for any composition ...
-- Reduce Eu3 Tl1 over-investment: Cap at 5--6 queries per Eu--Tl binary/ternary;
 this composition consumed 11 queries with marginal returns after the 8th stable
 result.
```

Input: `evaluation_history` at step 31 (BY COMPOSITION block).

```
Queries: 31 | Stable: 28/31 | Novel: 26 | Avg e_above_hull: 0.035 | Best: 0.000
Eu3 Tl2    : 8q, 7 stable, 7 novel, best=0.000 worst=0.127
Sn1 Tl4    : 9q, 9 stable, 9 novel, best=0.000 worst=0.051
Eu2 Sn1 Tl1: 9q, 9 stable, 5 novel, best=0.000 worst=0.018
Eu1 Sn1 Tl2: 2q, 1 stable, 2 novel, best=0.000 worst=0.107
Eu2 Tl1    : 1q, 1 stable, 1 novel, e=0.000
Eu2 Sn1 Tl2: 1q, 1 stable, 1 novel, e=0.099
Eu3 Sn1 Tl1: 1q, 0 stable, 1 novel, e=0.147
```

Output: Actor's next_thought.

> *". . .Looking at the evaluation history, Sn1Tl4 (Tl4Sn) has an exceptional 100% success rate (9/9 stable, 9 novel), which is even better than Eu3Tl2's 87.5%. The buffer has 2 Tl4Sn structures that haven't been evaluated yet.* ***Given the prior reflection guidance to limit queries per composition*** *and prioritize Tl-rich ternaries, I should move to this proven composition. I'll select a Tl4Sn structure for oracle evaluation to continue exploiting this highly successful composition region."*

Output: Actor's action.

```
next_tool_name: select_for_evaluation
next_tool_args: {''composition'': ''Tl4Sn'', ''structure_index'': 0,
    ''reason'': ''Sn1Tl4 has 100% success rate (9/9 stable, 9 novel) ...;
    prioritizing proven Tl-rich ternary composition''}
```

Failure to follow advice. At the moment of the decision, the actor's prompt (a) contains an explicit rule saying "*Cap at 2–3 queries per composition*", (b) shows Sn1 Tl4 already sitting at 9 queries in the current episode — more than three times the advised cap, and (c) is followed by a thought that *literally cites* "the prior reflection guidance to limit queries per composition" before recommending a 10th query on the same composition. The same pattern recurs at steps 15, 19, 21, 24, 28, 29, 33, 34, and 35 of this episode alone, covering Tl4Sn, Eu3Tl2, and Eu2TlSn, each of which is re-queried well after passing the 2–3 query cap. This rules out observability or memory as explanations: the actor sees the advice, sees its own per-composition query counts, sometimes even quotes the rule in its reasoning, and then violates it anyway. What we observe is not forgotten advice but ignored advice, with exploitation of short-term per-query reward overriding the cross-episode strategy that the reflector has encoded.