Evaluating LLMs as Constructive Editors: A Sequential Decision-Making Framework across Lexical, Compositional, and Biological Tasks

Anonymous ACL submission

Abstract

Many evaluations of large language models (LLMs) as sequential agents mix instruction-following errors with deeper planning failures. We study *constructive structure editing*, a class of tasks in which an agent iteratively modifies a partial structure through local edits so that the final artifact satisfies delayed global objectives and constraints. We instantiate this framework in three environments: lexical editing (Word Ladder), compositional editing (alloy design), and biological sequence editing (GB1-based design). Across prompting and edit-history conditions, we find that format-only support is weak, task-specific interventions help only when they match the dominant structural bottleneck, and recent edit history is useful only in environments that reward cumulative state tracking. Across all three tasks, the strongest persistent failures are structural rather than syntactic, including budget-unaware continuation, objective-tradeoff collapse, and epistasis-driven local-optimum traps. These results suggest that valid local editing does not imply robust constructive planning, and that sequential evaluation should test whether a model can build coherent structures through dependent edits, not only emit valid next actions.

1 Introduction

Large language models are increasingly evaluated as agents that act over multiple steps (Liu et al., 2023; Zhou et al., 2023; Wang et al., 2022b; Shridhar et al., 2021; Xie et al., 2024; Deng et al., 2023; Yao et al., 2022). Yet many sequential tasks are not only about selecting the next admissible action. They require the agent to *construct* an artifact by repeatedly editing a partial structure: a word path, a material composition, a biological sequence, a program, or a plan. In such problems, success depends not only on local action validity, but on whether early edits preserve the feasibility and usefulness of later edits.

This distinction matters because broad agent evaluations often make it hard to separate instruction-following failures from deeper trajectory-level planning failures (Liu et al., 2023; Zhou et al., 2023; Wang et al., 2022b; Xie et al., 2024; Paglieri et al., 2025; Yao et al., 2023). When tasks are broad and heterogeneous, weak performance can reflect many factors at once: instruction instability, missing tool knowledge, domain knowledge gaps, or deeper path-construction failures. Recent work has also noted that relying on final success alone can obscure which decision module is failing and where the relevant bottleneck lies (Li et al., 2024). We argue that constructive editing provides a useful middle ground: a common abstraction that is narrow enough for controlled failure analysis, but broad enough to support cross-domain comparison.

We study this setting through **constructive structure editing**. In this class of problems, an agent incrementally modifies a partial structure through local edits so that the final artifact satisfies global goals and constraints. These tasks are sequential not merely because they contain multiple turns, but because early edits change the meaning, feasibility, and value of later edits. A model can therefore fail even when each individual edit appears locally reasonable.

To operationalize this idea, we build three environments with the same interaction template: the agent observes a *partial structure*, executes a *local edit*, and may call `FINALIZE` to obtain delayed global evaluation. The three environments instantiate this template over different structure types: **Word Ladder** as lexical editing, **Alloy Design** as compositional editing, and **Biological Sequence Design** as sequence editing grounded in a GB1-based fitness landscape. Figure 1 summarizes this shared abstraction. The environments were chosen not as a strict complexity ladder, but to cover different sources of sequential difficulty: lexical

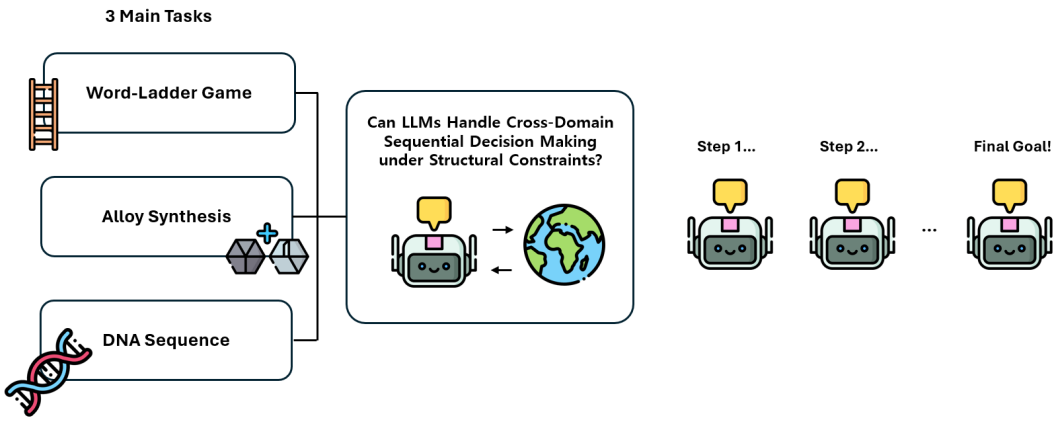

Figure 1: **Constructive structure editing as a shared sequential decision abstraction.** Across all three environments, the agent repeatedly edits a partial structure and receives delayed global evaluation only after finalization. The environments differ in structure type—lexical string, alloy composition, biological sequence—but share the same core challenge: early local edits alter the feasibility and utility of later edits.

path maintenance in Word Ladder, delayed multi-objective trade-offs in alloy design, and interaction-sensitive combinatorial search in GB1 sequence design. Together, they span a useful range from locally legible symbolic constraints to settings in which the relevant interaction structure is less explicit from the current state.

The environments also differ in how much of the relevant structure is visible from the current state. In Word Ladder, the main burden is symbolic path maintenance under explicit lexical constraints. In alloy design and GB1 sequence design, by contrast, the effect of a local edit is not captured by immediate plausibility alone. Alloy editing stands in for inverse-design settings in which composition changes must be judged through delayed property trade-offs and recoverability under budgeted search (Sánchez-Lengeling and Aspuru-Guzik, 2018; Wang et al., 2022a; Du et al., 2023). In GB1, locally appealing edits can still be misleading because fitness depends on higher-order epistatic interactions across sites and because protein fitness landscapes can be rugged even when local improvements appear accessible (Dallago et al., 2021; Sandhu et al., 2025). This difference provides one way to interpret the environments: as the structure governing success becomes less explicit in the current state, effective behavior may depend less on local validity alone and more on tracking delayed consequences and future possibilities (Ha and Schmidhuber, 2018; Sobal et al., 2025).

This framing helps separate surface action validity from structural planning. If a task is reducible to format compliance or candidate ranking, then weak performance may reveal little more than instruction brittleness. By contrast, constructive editing requires the model to decide *where* to edit, *how* to edit, *when* to stop, and *how* early edits constrain later ones. We therefore use lightweight surrogate evaluators, distinguish surface failures from structural failures, and test prompting and edit-history access to examine which failures are prompt-sensitive and which remain after local action validity improves.

This framing leads to three empirical questions. Does constructive editing expose more than action-format difficulty? Do prompt interventions help by improving syntax handling, or by changing how models manage edit trajectories? And when does recent trajectory access support construction rather than lock a model into a poor local path?

Contributions. We make three contributions. First, we formalize *constructive structure editing* as a practically important subclass of sequential decision making for LLMs, in which local edits transform a partial structure under delayed global evaluation. Second, we introduce a unified three-environment framework spanning lexical, compositional, and biological sequence editing under a shared interaction protocol. Third, through prompting and history ablations plus failure analysis, we show that format-only support is weak, prompt gains are environment-dependent, recent edit history is only conditionally helpful, and persistent

failures are structural rather than syntactic.

2 Related Work

LLMs as sequential agents. Recent benchmarks evaluate LLMs in embodied, web-based, game-based, computer-use, and scientific settings (Liu et al., 2023; Zhou et al., 2023; Wang et al., 2022b; Shridhar et al., 2021; Deng et al., 2023; Yao et al., 2022; Xie et al., 2024; Paglieri et al., 2025). These benchmarks show that multi-step interaction is challenging, but often mix many sources of difficulty. Recent work has also emphasized the value of decomposing agent behavior rather than relying only on end-task success (Li et al., 2024). Our work complements these benchmarks by focusing on a narrower class of tasks whose shared difficulty is incremental construction under delayed evaluation.

Constructive search, editing, and scientific design. There is a long tradition of viewing output construction as search over partial or structured outputs. Structured prediction has been cast as search or MDP-style decision making, where policies iteratively construct outputs (Doppa et al., 2014). Related ideas also appear in iterative editing and refinement models for text generation and transformation (Guu et al., 2018; Malmi et al., 2019; Gu et al., 2019). In scientific design, local edits often interact through structure that is only partially visible from the current state. In materials science, inverse design optimizes target properties under coupled relations among composition, structure, processing, and performance (Sánchez-Lengeling and Aspuru-Guzik, 2018; Wang et al., 2022a; Borg et al., 2020; Du et al., 2023). In protein engineering, sequence optimization takes place over rugged fitness landscapes shaped by non-additive mutational interactions (Dallago et al., 2021; Sandhu et al., 2025). Related work on world models and model-based planning likewise emphasizes tracking how actions shape future states (Ha and Schmidhuber, 2018; Sobal et al., 2025). Our benchmark draws on these ideas not to reproduce full domain realism, but to study controlled editing tasks with different degrees of latent interaction structure.

Positioning of this work. Our benchmark is not intended to replace broad agent evaluations such as AGENTBENCH, WEBARENA, or OSWORLD (Liu et al., 2023; Zhou et al., 2023; Xie et al., 2024). Instead, it targets a narrower diagnostic question: whether a model can iteratively build a coherent artifact through dependent local edits when success is revealed only under delayed global evaluation. The three environments should therefore be read as contrasting instantiations of the same constructive-editing protocol with different dominant structural bottlenecks. This narrower scope enables more controlled attribution of failure to syntax, trajectory management, and delayed structural constraints.

3 Method

Our goal is to evaluate a specific subclass of sequential decision problems: *constructive structure editing*. In this setting, the agent does not choose among fixed candidates. It iteratively transforms a partial structure through local edits so that the final artifact satisfies global objectives and constraints. This framing is related in spirit to earlier views of structured prediction as search over partial or complete outputs (Doppa et al., 2014).

3.1 Constructive Structure Editing as a Sequential Decision Problem

We model each environment $e \in \mathcal{E}$ as

$$\mathcal{T}_e = (\mathcal{S}_e, \mathcal{A}_e, P_e, O_e, G_e, H_e),$$

where $\mathcal{S}_e$ is the space of partial structures, $\mathcal{A}_e$ is the action space of local edits plus `FINALIZE`, P_e is the transition dynamics induced by edits, O_e is the observation function, G_e is the delayed global evaluator, and H_e is the edit budget.

At step t, the agent receives

$$o_t = O_e(x_t, \tau_{<t}),$$

where $x_t \in \mathcal{S}_e$ is the current partial structure and $\tau_{<t}$ is the preceding edit trajectory. The agent then outputs either a local edit

$$a_t \in \mathcal{A}_e^{\text{edit}}$$

or `FINALIZE`. A local edit updates the structure by

$$x_{t+1} = P_e(x_t, a_t),$$

while `FINALIZE` triggers delayed evaluation

$$y = G_e(\tau_{\leq t}, x_t).$$

This differs from one-step prediction in two ways. First, the partial structure is semantically meaningful: a partially edited word, composition, or sequence changes what later edits are useful, legal, or recoverable. Second, success may depend

on the final state and on path-dependent quantities induced by the trajectory. We therefore write the success predicate as

$$S(\tau, x_T) \in \{0, 1\},$$

and define success rate under model m, prompt condition p, and history condition k as

$$\mathrm{SR}(e, m, p, k) = \mathbb{E}_{\tau \sim \pi_m^{(p,k)}, \mathcal{T}_e} \left[S(\tau, x_T) \right].$$

3.2 What makes the task genuinely sequential?

A task is not informative as sequential decision making merely because it contains multiple turns. We require each environment to satisfy four properties: (i) **local editability**, so the problem does not collapse into one-shot generation; (ii) **path dependence**, so early edits alter later feasibility or utility; (iii) **partial-state semantics**, so intermediate states are meaningful; and (iv) **delayed global evaluation**, so final success is resolved only after `FINALIZE`. In practice, all three environments also exhibit **order sensitivity**: changing edit order affects repair cost, structural compatibility, or final evaluability even when the high-level objective is unchanged.

These criteria distinguish constructive editing from tasks that are merely multi-turn on the surface. A benchmark can contain several turns while still being reducible to repeated format compliance or local candidate choice. Our environments are designed to avoid that failure mode. The intended challenge is not only choosing a valid edit, but sustaining a coherent construction trajectory.

3.3 Environment Family

We instantiate this formulation over three structure types: **lexical strings**, **alloy compositions**, and **biological sequences**. All environments share the same interaction skeleton: the model observes a partial structure, performs a local edit under an explicit edit grammar, receives limited local feedback, and decides when to stop and request delayed global evaluation. The environments are joined by this shared protocol rather than by a claim of strict domain progression: they were selected to cover different dominant sources of constructive difficulty under an otherwise matched interface.

Word Ladder as lexical editing. A state contains the current word w_t, the target word $w^\star$, the set of visited words, and the remaining edit budget. The model outputs

$$\texttt{EDIT}\ \langle i \rangle\ \langle c \rangle,$$

where i is a character position and c is a replacement letter. An edit is valid iff the resulting word exists in the lexicon, differs by exactly one character, and has not already been visited. The task succeeds when the target word is reached. This is the minimal control task, where the main challenge is path maintenance rather than domain knowledge.

Alloy design as compositional editing. A state is a composition vector over six elements with total mass fixed at 100. The model outputs

$$\texttt{EDIT}\ \langle e^+ \rangle\ \langle e^- \rangle\ \langle \Delta \rangle,$$

which increases one element and decreases another by the same amount. The first half of the budget allows coarse edits and the second half allows finer adjustments. Intermediate observations expose estimated UTS, exact density, and lightweight trajectory signals, with the UTS surrogate grounded in an MPEA mechanical properties dataset (Borg et al., 2020). Final success is determined at `FINALIZE` by combining final-property thresholds with path-sensitive penalties such as recovery-cost growth. The dominant bottleneck is sustained management of a delayed UTS–density trade-off.

GB1 sequence design as biological editing. A state is a short sequence with estimated fitness, stability status, and remaining budget. The model outputs

$$\texttt{EDIT}\ \langle i \rangle\ \langle v \rangle,$$

which replaces the token at position i with value v. Intermediate feedback includes additive single-site estimates, but final success depends on delayed evaluation under a combinatorial landscape with epistatic interactions grounded in GB1 fitness data as curated in FLIP (Dallago et al., 2021). This makes the task sensitive to misleading local gains and creates local optima that cannot be solved by additive reasoning alone.

3.4 Prompting and History Conditions

We test six lightweight prompting conditions: **zero-shot**, **few-shot (format-only)**, **few-shot (strategy-bearing)**, **scaffold**, **self-check (generic)**, and **self-check (task-specific)**. All conditions preserve the same action-only output format. They therefore

differ mainly in the behavioral bias they impose on the editing policy, not in the freedom to generate additional reasoning text.

This distinction matters for interpretation. If prompt improvements arose mainly from better schema compliance, then format-only demonstrations should have helped broadly. If instead performance changes only when the prompt alters how the model approaches the edit sequence, then prompt effects are better understood as policy shaping. Our intervention design is intended to separate these two possibilities.

We also vary recent edit-history access: $M0$ shows only the current structure, $M1$ adds the most recent edit, $M3$ adds the most recent three edits. This is not meant as a generic memory benchmark. It specifically tests whether recent trajectory access helps the model maintain a constructive path rather than merely remember what it last did.

3.5 Baselines, Calibration, and Failure Analysis

We compare LLM behavior with three baseline families: **random**, **immediate-greedy**, and a **sequence-aware heuristic**. The purpose is diagnostic. If sequence-aware planning clearly outperforms myopic local optimization, then the environment exhibits genuine path-dependent structure, in the spirit of search-based approaches to structured construction (Doppa et al., 2014). If not, low LLM performance would be harder to interpret.

Before the main sweeps, we calibrate each environment so that random remains poor, immediate-greedy is better but still limited, and a sequence-aware baseline performs best. This prevents the suite from being either trivial or dominated by action-format confounds.

Finally, we separate **surface edit failures** from **structural planning failures**. Surface failures include malformed or locally invalid actions. Structural failures include budget-unaware continuation, premature finalization, objective-tradeoff collapse, recovery-cost explosion, and local-optimum traps. This taxonomy is the main mechanism by which we operationalize the distinction between valid action emission and constructive planning.

4 Experimental Setup

We evaluate a controlled family of non-thinking instruction models under matched prompting and history conditions. Our goal is not broad leaderboard coverage, but controlled analysis of constructive failure patterns under a fixed API-based inference setup.

Main prompt experiment. We run six intervention conditions at $M0$: zero-shot, few-shot (format-only), few-shot (strategy-bearing), scaffold, self-check (generic), and self-check (task-specific).

History ablation. We evaluate selected prompt conditions at $M0$, $M1$, and $M3$. This isolates the effect of recent trajectory access from the effect of prompt intervention itself.

Calibration and robustness packages. We additionally run baseline calibration sweeps, an alloy exemplar-robustness study, GB1 operating-point validation, and a word-length ablation for the lexical control task.

Scale. Across all packages, the final study contains roughly 3,160 episodes under a single consistent code base.

5 Results

We organize the empirical analysis around three issues. First, we test whether the benchmark exposes more than surface action-format difficulty. Second, we examine whether prompt interventions affect only local compliance or instead reshape the editing policy itself. Third, we ask when recent edit history helps constructive performance rather than reinforcing a poor local trajectory. We then use calibration and robustness analyses to verify that the observed patterns reflect meaningful path-dependent structure rather than task degeneracy.

5.1 The benchmark exposes more than action-format difficulty

Table 1 provides the clearest evidence against a pure format-compliance interpretation. If performance were mainly limited by action syntax, then few-shot prompting with format demonstrations should improve performance broadly. Instead, the format-only condition underperforms zero-shot in all three environments: Word Ladder, alloy editing, and GB1 sequence editing.

This pattern narrows the interpretation of failure. The environments do require strict action grammars, but better schema exposure alone does not resolve the main difficulty. The benchmark therefore appears to measure more than whether the model can produce a well-formed next action. It

also measures whether the model can maintain a useful construction path under delayed evaluation.

5.2 Prompt interventions help only when they align with structural demands

Table 1 also shows that prompt interventions are not uniformly helpful. Performance changes only when the prompt changes how the model approaches the edit sequence, and the effect depends strongly on the environment.

In Word Ladder, the best condition is the scaffold. This is consistent with a control task where legality checking, progress monitoring, and stopping decisions are central. In alloy editing, most interventions are harmful, and only task-specific self-checking recovers zero-shot performance. This suggests that the main challenge is sustained management of a delayed UTS–density trade-off rather than local syntax. In GB1, task-specific self-checking performs best, but the gain remains well below the privileged sequence-aware upper bound. This indicates that prompt engineering can help, but does not remove the underlying interaction-sensitive bottleneck.

Taken together, these results support a narrower interpretation of prompting. In this benchmark, prompts behave less like general assistance mechanisms and more like lightweight policy priors. When the prior aligns with the dominant structural demand of the environment, performance can improve. When it does not, the effect is weak or even harmful.

5.3 Residual failures are structural and differ by environment

Success rates alone do not show what kind of constructive difficulty remains. To interpret the benchmark more precisely, we analyze the dominant failure mode in each environment.

Word Ladder is primarily limited by BUDGET-UNAWARE ACTION: the model often continues after entering a low-value path from which recovery is unlikely. Alloy editing is dominated by OBJECTIVE-TRADEOFF FAILURE, where local edits improve one objective while making the final UTS–density pair infeasible. GB1 sequence editing is dominated by LOCAL-OPTIMUM TRAP, often followed by PREMATURE FINALIZE: the model overtrusts misleading local signals and fails to make the globally necessary edit.

These patterns matter for two reasons. First, they match the intended bottleneck of each environment, which supports the design of the benchmark. Second, they show that the main residual difficulty is not uniform across domains. The suite does not expose one generic multi-step weakness. It exposes different ways in which constructive planning can fail.

GB1 is especially informative because surface invalidity is nearly absent under most prompt conditions, yet structural failure remains common. Once legal edit emission is mostly solved, the dominant remaining problem is the inability to escape a deceptive local trajectory. This cross-environment contrast is also consistent with the fact that the relevant structure is more explicit in Word Ladder and less locally legible in alloy and GB1.

5.4 Recent edit history helps only when trajectories carry useful cumulative signal

Table 2 shows that recent edit history is not a uniformly beneficial resource. Its value depends on whether the current trajectory contains useful cumulative signal or misleading local evidence.

In alloy editing, history helps under zero-shot prompting: success rises from 12% at $M0$ to 25% at $M3$. This is consistent with a task where the model benefits from tracking how earlier edits shaped a delayed multi-objective trade-off. By contrast, history is harmful in Word Ladder. The strongest lexical condition, scaffold, drops from 30% at $M0$ to 10% at $M3$, suggesting that recent trajectory access can reinforce commitment to a dead-end path.

The strongest negative history effect appears in GB1. Under task-specific self-check prompting, success drops from 35% at $M0$ to 5% at both $M1$ and $M3$. The analogous zero-shot condition drops from 25% to 8%. In this setting, recent history appears to preserve the wrong local rationale rather than correct it.

This result sharpens the difference between *remembering a trajectory* and *using a trajectory well*. In constructive editing, memory is helpful only when the trajectory itself is a good guide for future edits.

5.5 Baseline comparisons confirm genuine path dependence

The baseline hierarchy helps validate the benchmark itself. In alloy editing, random performs poorly, myopic greedy baselines improve slightly, and stronger sequence-aware planning performs best. In GB1, additive local heuristics fail across

Table 1: Main prompt-intervention results at $M0$. Values are success rates (SR, %) under the final code version. Format-only few-shot prompting does not reliably improve performance over zero-shot, while larger gains appear only when the intervention changes the model's editing policy in a way that matches the dominant structural bottleneck of the environment. The best condition therefore differs across environments rather than transferring uniformly.

Environment	Zero-shot	Few-shot (format)	Few-shot (strategy)	Scaffold	Self-check (generic)	Self-check (task-specific)
Word Ladder (4-letter)	17	13	20	**30**	3	13
Alloy editing	**20**	13	3	13	3	**20**
GB1 sequence editing	22	20	24	12	26	**38**

Table 2: History sensitivity under selected prompt conditions. Values are success rates (SR, %) under the final code version. Recent edit history is not uniformly beneficial: it helps in alloy editing, where cumulative trajectory information supports trade-off tracking, but harms performance in Word Ladder and GB1, where additional history can reinforce commitment to unproductive local paths.

Environment	Prompt	$M0$	$M1$	$M3$	Interpretation
Word Ladder	zero-shot	22	20	10	history is mildly harmful
Word Ladder	scaffold	30	18	10	history strongly harms path construction
Alloy	zero-shot	12	20	25	history helps cumulative trade-off tracking
Alloy	self-check	18	8	18	non-monotonic history effect
GB1	zero-shot	25	8	8	history strongly harms sequence construction
GB1	self-check	35	5	5	catastrophic degradation under history

threshold settings, while the LLM and a privileged noisy oracle remain substantially above zero.

These comparisons matter because they rule out two simpler interpretations. The tasks are not trivial local optimization problems, since one-step heuristics fail. They are also not degenerate impossible tasks, since sequence-aware policies and privileged baselines perform much better. The observed LLM failures therefore arise in environments where path-aware behavior is measurably useful.

5.6 Summary of empirical findings

Across the full study, four empirical conclusions are stable. First, the benchmark is not primarily measuring action-format compliance. Second, prompt interventions matter mainly when they reshape the editing policy, and their effect is strongly environment-dependent. Third, recent edit history is conditionally useful rather than uniformly helpful. Fourth, the strongest persistent failures are structural rather than syntactic.

Taken together, the results support the paper's central empirical claim: LLMs can often produce valid local edits, yet still fail to construct coherent final structures through sequences of dependent edits. This is the core diagnostic value of the benchmark.

6 Discussion

The main implication of our findings is diagnostic: valid next-action generation and coherent artifact construction should not be treated as the same capability. Across all three environments, models often produce locally valid edits while still failing to preserve future feasibility over the full trajectory.

Three contributions follow from this result. **First**, the benchmark shows that broad sequential failure can be decomposed more precisely. Once tasks are recast as constructive editing under delayed evaluation, it becomes possible to distinguish

surface action validity from structural planning failure, and to attribute errors more directly to syntax, trajectory management, and delayed constraints. This is the main value of the benchmark as an evaluation lens.

Second, the three environments provide a useful contrast in the kind of structure that planning must respect. Word Ladder is governed by explicit symbolic constraints, so the main challenge is path maintenance under a budget. In alloy editing and GB1 sequence design, by contrast, local edits must be judged against interactions that are not fully revealed by the immediate observation. In alloy, useful edits must be evaluated in light of delayed trade-offs and recoverability (Sánchez-Lengeling and Aspuru-Guzik, 2018; Wang et al., 2022a; Du et al., 2023). In GB1, an edit with positive local evidence may still move the trajectory toward a poor basin because mutational effects do not compose additively and because rugged landscapes can make local improvement systematically deceptive (Dallago et al., 2021; Sandhu et al., 2025). This is an interpretation of the environments, not a claim that the benchmark fully models physical or biological mechanism. Its value is comparative: the shared protocol makes it possible to contrast constructive planning under relatively explicit symbolic constraints with planning under more latent interaction structure.

Third, the prompt and history results clarify what kinds of support help in constructive settings. The weak performance of format-only prompting argues against a simple action-syntax account. If the main barrier were action format, then explicit format demonstrations should have helped more consistently. Instead, improvements appear only when the prompt changes how the model manages the edit trajectory. In that sense, prompts behave less like generic assistance and more like lightweight policy priors whose value depends on the dominant structural bottleneck of the environment. The history ablation points in the same direction. Recent edit history is not a uniformly beneficial memory resource: in alloy editing it improves performance because the task rewards cumulative trade-off tracking, whereas in Word Ladder and GB1 the same information can strengthen commitment to an unproductive path. A model can therefore remember what it did without becoming better at planning what it should do next.

More broadly, these findings help clarify the link between constructive editing and internal modeling. As the structure governing success becomes less explicit in the local state, planning may depend less on one-step validity and more on tracking how edits change future possibilities (Ha and Schmidhuber, 2018; Sobal et al., 2025). We do not claim that these experiments directly measure a learned world model. Rather, the results suggest a narrower point: controlled constructive environments can expose cases where local edit validity is largely solved while trajectory quality still depends on an implicit model of delayed consequences.

This interpretation also sharpens the paper's boundary conditions. We do not claim that all sequential decision problems should be reduced to constructive editing. Many important agent tasks involve tool use, exploration, dialogue management, or external state estimation that are not captured here. Nor do we claim that failure in alloy or GB1 should be read as failure on full scientific reasoning. Yet these narrower environments reveal something broad benchmarks can hide: the inability to maintain a coherent build trajectory even when local actions look valid.

7 Conclusion

We proposed constructive structure editing as a controlled evaluation lens for sequential settings in which success depends on building, not only acting. Rather than asking whether a model can simply select an admissible next action, we evaluate whether it can iteratively transform a partial structure through local edits while preserving the feasibility and utility of future edits.

Across Word Ladder, alloy composition, and biological sequence design, the main empirical picture is consistent: format-only support is weak, prompt interventions act mainly by changing editing policy, recent trajectory access is only conditionally useful, and persistent failures are predominantly structural rather than syntactic. Valid local editing does not guarantee coherent multi-step construction.

The framework is intended not as a leaderboard, but as a diagnostic tool for identifying when models can emit legal edits yet still fail to construct a coherent final artifact. More broadly, the results suggest that sequential evaluation should test not only whether models can act validly at each step, but whether they can preserve future feasibility while building toward a final structure.

8 Limitations

Constructive abstraction versus full realism. Our environments intentionally abstract away much of the complexity of real scientific design. Alloy development involves thermodynamics, kinetics, processing conditions, and microstructure, while biological sequence design involves richer structural and experimental constraints than those represented here. We therefore position the benchmark as an evaluation of *constructive editing structure*, not of full domain realism.

Surrogate–reality gap. Alloy and biological sequence evaluation rely on surrogate or semi-synthetic scorers rather than full simulation or wet-lab validation. This is necessary for controlled large-scale evaluation, but it limits claims about scientific deployment.

Restricted edit grammars. Although the environments do not reduce to menu selection, they still use constrained edit grammars to avoid confounding constructive planning with arbitrary free-form generation. This improves interpretability, but underrepresents fully open design spaces.

Scale of edit horizons. The edit budgets are intentionally modest to keep evaluation reproducible under API-based inference. These budgets are sufficient to induce path dependence, but they remain shorter than many real design workflows.

Exemplar sensitivity and model scope. The alloy robustness ablation shows that strategy-bearing demonstrations can be highly sensitive to the exact trajectory shown. In addition, our controlled study uses one model family, which improves causal interpretation but limits generalization across providers and alignment regimes. Accordingly, our claims concern controlled constructive planning burdens rather than a universal account of all LLMs or full physical or biological mechanism modeling.

Acknowledgments

References

Christopher K. H. Borg, Carolina Frey, Jasper Moh, Tresa M. Pollock, Stéphane Gorsse, Daniel B. Miracle, Oleg N. Senkov, Bryce Meredig, and James E. Saal. 2020. Expanded dataset of mechanical properties and observed phases of multi-principal element alloys. *Scientific Data*, 7(1):430.

Christian Dallago, Jody Mou, Kadina Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Emmanuele Papa, Sabina Shakhimardanova, Alex Cornett, Johannes Welbl, Aleksandr Shtedritski, Boris Burkov, Ziqing Yang, Axel X. Lu, Anastasiia Kozlova, Sungroh Yoon, Alaa Nawy, Sarah Min, Steven E. Brenner, and 18 others. 2021. FLIP: Benchmark tasks in fitness landscape inference for proteins. *NeurIPS Datasets and Benchmarks*, 1.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *arXiv preprint arXiv:2306.06070*.

Janardhan Rao Doppa, Alan Fern, and Prasad Tadepalli. 2014. Structured prediction via output space search: The search-based structured prediction framework. *Journal of Machine Learning Research*, 15:1317–1350.

Yuanqi Du, Yingheng Wang, Yining Huang, Jianan Canal Li, Yanqiao Zhu, Tian Xie, Chenru Duan, John M. Gregoire, and Carla P. Gomes. 2023. M^2hub: Unlocking the potential of machine learning for materials discovery. In *Advances in Neural Information Processing Systems*.

Jiatao Gu, Changhan Wang, Jake Zhao, and 1 others. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122*.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. 2024. Embodied agent interface: Benchmarking LLMs for embodied decision making. In *Advances in Neural Information Processing Systems*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2023. Agentbench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065.

Marco Paglieri and 1 others. 2025. BALROG: Benchmarking agentic LLM and VLM reasoning on games. *arXiv preprint arXiv:2501.XXXXX*.

Benjamín Sánchez-Lengeling and Alan Aspuru-Guzik. 2018. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365.

Mahakaran Sandhu, John Chen, Dana Matthews, Matthew A. Spence, Sacha B. Pulsford, Barnabas Gall, James Nichols, Nobuhiko Tokuriki, and Colin J. Jackson. 2025. Computational and experimental exploration of protein fitness landscapes: Navigating smooth and rugged terrains. *Biochemistry*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*.

Vlad Sobal, Wancong Zhang, Kynghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. 2025. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*.

Jia Wang, Yingxue Wang, and Yanan Chen. 2022a. Inverse design of materials by machine learning. *Materials*, 15(5):1811.

Ruoyao Wang, Akihiro Matsumoto, Xiaohan Wu, Aniruddha Kembhavi, Takumi Ito, Hannaneh Hajishirzi, Roozbeh Mottaghi, and Ali Farhadi. 2022b. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11303.

Tianbao Xie, Yang Zou, Peiyi Zhang, Boyuan Deng, Qinsheng Wang, Krishna Kodur, Yucheng Gu, Yiyuan Zha, Chuxin Chen, Zhenlong Xu, Jianfeng Zhou, Yiheng An, Jun Chen, Tao Yin, Yilun Wang, Chunxiang Xiong, David Hsu, Enze Xie, Ping Luo, and 3 others. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Appendix

This appendix provides supporting formalization, environment-construction details, detailed results tables, representative failure patterns, and abbreviated prompt templates.

A Methodological Formalization

This appendix formalizes the constructive editing framework used in the main paper. Our goal is not to redefine general sequential decision making, but to specify the narrower class of problems studied here: *constructive structure editing*. This formalization clarifies what is shared across lexical, compositional, and sequence editing environments and makes explicit which quantities are measured in the empirical study. It is related in spirit to search-based views of structured construction, in which outputs are built through traversal of a structured search space (Doppa et al., 2014).

A.1 Constructive Editing Task

We define a constructive editing task as a tuple

$$\mathcal{T} = (\mathcal{X}, \mathcal{A}, P, O, G, B, S),$$

where $\mathcal{X}$ is the space of partial structures, $\mathcal{A}$ is the action space of local edits together with a terminal action `FINALIZE`, P is the transition operator induced by a local edit, O is the observation function shown to the language model, G is the delayed global evaluator, $B \in \mathbb{N}$ is the maximum edit budget, and S is the binary success predicate.
At step t, the environment state is a partial structure

$$x_t \in \mathcal{X},$$

and the model receives an observation

$$o_t = O(x_t, \tau_{<t}),$$

where

$$\tau_{<t} = (x_0, a_0, x_1, a_1, \ldots, x_{t-1}, a_{t-1})$$

is the preceding edit trajectory.
The model outputs either a local edit

$$a_t \in \mathcal{A}_{\text{edit}}$$

or the terminal action

$$a_t = \texttt{FINALIZE}.$$

If $a_t \in \mathcal{A}_{\text{edit}}$, the next state is

$$x_{t+1} = P(x_t, a_t).$$

If $a_t = \texttt{FINALIZE}$, the episode terminates and the final artifact is evaluated by

$$y = G(\tau_{\leq t}, x_t),$$

where $\tau_{\leq t}$ includes the full edit history up to termination.
The episode succeeds iff

$$S(\tau_{\leq t}, x_t) = 1.$$

A.2 Why This Is a Sequential Decision Problem

The tasks studied here are sequential in a stronger sense than simple repeated schema-following. We require four properties.

Local editability. Each action changes only a small part of the current structure.

Partial-state semantics. Intermediate states are meaningful objects that constrain future edits.

Path dependence. The value of a local edit depends not only on the current state, but also on the trajectory by which that state was reached.

Delayed global evaluation. Intermediate observations may expose proxy signals, but the final success predicate is resolved only after `FINALIZE`.
Together, these properties define the specific notion of sequential decision making studied in this paper.

A.3 Environment-Specific Instantiations

The three environments differ in state and action spaces, but all instantiate the same template.

Lexical editing. The state is a word w_t together with a target word $w^\star$, a visited-word set, and a remaining budget. An action is

$$a_t = \texttt{EDIT}(i, c),$$

which replaces character position i with character c. Validity requires that the resulting word belong to the lexicon, differ by exactly one character, and not appear in the previously visited set.

Alloy editing. The state is a composition vector

$$x_t = (c_{\mathrm{Fe}}, c_{\mathrm{Ni}}, c_{\mathrm{Cr}}, c_{\mathrm{Co}}, c_{\mathrm{Mn}}, c_{\mathrm{Mo}})$$

with

$$\sum_j c_j = 100.$$

An action is

$$a_t = \texttt{EDIT}(e^+, e^-, \Delta),$$

which increases one element and decreases another by the same amount.

GB1 sequence editing. The state is a short editable sequence

$$x_t = (r_1, r_2, \ldots, r_L),$$

where each r_i is a token at editable site i. An action is

$$a_t = \texttt{EDIT}(i, v),$$

which replaces token r_i with v.

A.4 Prompted Policy

Let $\pi_\theta^{(p,k)}$ denote the policy induced by model θ under prompt condition p and history condition k. At each step,

$$a_t \sim \pi_\theta^{(p,k)}(\cdot \mid o_t),$$

where o_t already contains the effect of the chosen prompt template and the available recent history.
The prompt condition

$$p \in \{\text{zero-shot}, \text{few-shot-format}, \text{few-shot-strategy}, \text{scaffold}, \text{self-check-generic}, \text{self-check-task}\}$$

determines the behavioral intervention. The history condition

$$k \in \{M0, M1, M3\}$$

determines how much recent edit history is included in the observation.

A.5 Success Rate, Prompt Effect, and History Effect

For a task $\mathcal{T}$ and a prompted policy $\pi_\theta^{(p,k)}$, the primary evaluation metric is success rate,

$$\mathrm{SR}(\theta, p, k; \mathcal{T}) = \mathbb{E}_{\tau \sim \pi_\theta^{(p,k)}, \mathcal{T}} \left[S(\tau_{\leq T}, x_T) \right].$$

In practice, this quantity is estimated over N episodes as

$$\widehat{\mathrm{SR}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1} \left[S(\tau_{\leq T}^{(n)}, x_T^{(n)}) = 1 \right].$$

Prompt effects are measured relative to zero-shot at fixed history access:

$$\Delta_{\text{prompt}}(p; \mathcal{T}) = \mathrm{SR}(\theta, p, M0; \mathcal{T}) - \mathrm{SR}(\theta, \text{zero-shot}, M0; \mathcal{T}).$$

We further distinguish the format-only effect,

$$\Delta_{\text{format}} = \mathrm{SR}(\theta, \text{few-shot-format}, M0; \mathcal{T}) - \mathrm{SR}(\theta, \text{zero-shot}, M0; \mathcal{T}),$$

from the strategy-bearing effect,

$$\Delta_{\text{strategy}} = \mathrm{SR}(\theta, \text{few-shot-strategy}, M0; \mathcal{T}) - \mathrm{SR}(\theta, \text{zero-shot}, M0; \mathcal{T}).$$

History effects are measured relative to $M0$:

$$\Delta_{\text{hist}}(k; \mathcal{T}, p) = \mathrm{SR}(\theta, p, k; \mathcal{T}) - \mathrm{SR}(\theta, p, M0; \mathcal{T}), \qquad k \in \{M1, M3\}.$$

A.6 Failure Events

Let $\mathcal{F}$ denote the set of all logged failure events. For an episode τ, we record a multiset

$$F(\tau) = \{f_1, f_2, \ldots, f_m\}, \qquad f_i \in \mathcal{F}.$$

We partition $\mathcal{F}$ into three groups.

Surface edit failures.

$$\mathcal{F}_{\text{surface}} \subset \mathcal{F}$$

includes malformed actions, invalid positions, invalid tokens, and other schema-level failures.

Edit-process failures.

$$\mathcal{F}_{\text{process}} \subset \mathcal{F}$$

includes oscillation, exact undo loops, repeated dominated edits, and budget-unaware continuation.

Structural planning failures.

$$\mathcal{F}_{\text{struct}} \subset \mathcal{F}$$

includes premature finalization, objective-tradeoff collapse, recovery-cost explosion, local-optimum traps, and interaction-sensitive failures.

For a specific failure type $f \in \mathcal{F}$, we define its empirical frequency under environment $\mathcal{T}$, model θ, prompt p, and history condition k as

$$\widehat{P}(f \mid \theta, p, k; \mathcal{T}) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}[f \in F(\tau^{(n)})].$$

A.7 Baseline Hierarchy

Let

$$\Pi_{\text{rand}}, \quad \Pi_{\text{myopic}}, \quad \Pi_{\text{aware}}$$

denote the random, immediate-greedy, and sequence-aware baseline families.
The desired calibration regime is

$$\mathrm{SR}(\Pi_{\text{rand}}) < \mathrm{SR}(\Pi_{\text{myopic}}) < \mathrm{SR}(\Pi_{\text{aware}}).$$

This appendix-level formalization is intentionally narrower than a full MDP or POMDP treatment of all possible agent tasks. Its purpose is to make explicit what is shared across the three environments, what is being measured, and why the reported results support a structural rather than purely syntactic interpretation of LLM failure. It is also consistent with search-based perspectives in which structured outputs are constructed through sequential exploration rather than chosen in a single step (Doppa et al., 2014).

B Environment Construction Details

This appendix describes how the three evaluation environments were constructed and calibrated. Our goal was not to maximize domain realism, but to instantiate a shared constructive-editing protocol in which local edits, path dependence, and delayed global evaluation are all consequential. We document the design goals, instance construction procedures, operating-point selection, and filtering rules used in the final benchmark.

B.1 Design Goals

The environments were designed to satisfy five criteria:

1. **Irreducibility to one-shot generation.** Each environment requires a sequence of local edits rather than a single output, so that trajectory-level failures can emerge.
2. **Valid-but-suboptimal trajectories.** At every step the agent faces multiple valid actions, most of which lead to suboptimal or infeasible final states. Surface formatting correctness therefore does not predict task success.
3. **Baseline separability.** The operating regime was chosen so that random, observation-level greedy, and privileged oracle baselines occupy distinct performance tiers.
4. **Shared interaction protocol.** All three environments expose the same interface: a text observation, an `EDIT` / `FINALIZE` action grammar, per-step validity feedback, and a terminal success signal.
5. **Consequential delayed evaluation.** The globally evaluated quantity is revealed only upon `FINALIZE` or budget exhaustion, forcing the agent to plan under incomplete information.

B.2 Shared Interaction Protocol

Every environment implements the following loop:

1. **Reset.** A seed-dependent starting state is sampled and an initial observation is rendered as natural-language text.
2. **Step.** The agent produces a raw text action. The environment parses it (`EDIT ...` or `FINALIZE`), validates it against domain constraints, applies it if valid, and returns an updated observation together with any failure events.
3. **Termination.** The episode ends when the agent issues a valid `FINALIZE`, the step budget is exhausted, or a dead-end state is reached.
4. **Success criterion.** Evaluated at termination against the environment's hidden global objective.

The action grammar is environment-specific but always begins with `EDIT` followed by positional and value arguments. Invalid actions consume a step but do not alter the state. Table 3 summarizes the key dimensions of each environment.

Table 3: Environment summary. All environments share the same interaction protocol—text observation, `EDIT`/`FINALIZE` actions, local validity feedback, and delayed terminal evaluation— but differ in state representation, action grammar, and the dominant source of constructive difficulty. The table highlights why the benchmark is comparable across domains without collapsing them into the same task.

	Word Ladder	**Alloy**	**GB1 Protein**
Structure type	string (word)	composition vector	amino acid sequence
State dimension	4 characters	6 element percentages	4 residue positions
Action grammar	`EDIT pos letter`	`EDIT inc dec` δ	`EDIT pos aa`
Action space (approx.)	4×25	$30 \times \lvert\Delta\rvert$	4×4
Budget	$5 \times k_{\text{opt}}$	8 steps	6 steps
Evaluation instances	409 pairs	continuous (jittered)	625 variants
Observation noise	none	UTS $\pm$30 MPa	fitness $\pm$0.20
Key difficulty	dead-end avoidance	UTS–density tradeoff	epistasis trap

B.3 Word Ladder Instance Construction

Lexicon and graph. We use the Enable1 word list[1], a standard Scrabble tournament dictionary. After filtering to 4-letter words we obtain a vocabulary of 3903 words. We construct a Hamming-1 adjacency graph over this vocabulary ($|E| = 21\,002$); the largest connected component contains 3807 words (97.5%). The average degree is 9.9.

Pair selection. We sample 500 candidate start–target pairs from the largest connected component and compute BFS shortest-path lengths. We retain pairs with shortest-path distance $k_{\text{opt}} \in [3, 6]$, yielding **409 pairs**. The lower bound excludes trivial one- or two-step instances, and the upper bound excludes instances that require very long optimal paths.

Budget and observation. Each episode receives a budget of $\max(5 \times k_{\text{opt}},\ 15)$ steps. At each step the agent sees the target word, the current word with mismatch markers, the path taken so far, the remaining budget, and a list of valid replacement letters per position. The observation does *not* include BFS distance to the target.

Word length justification. We select 4-letter words as the main setting based on a difficulty ablation (Appendix I). Three-letter words are easier overall, while five-letter words are substantially sparser and more dead-end prone. Four-letter instances provide the most informative operating regime.

B.4 Alloy Environment Construction

Composition space. The agent edits a six-element alloy composition over Fe, Ni, Cr, Co, Mn, and Mo, subject to a simplex constraint ($\sum_i c_i = 100\%$) and per-element bounds.

Action grammar. Each `EDIT inc dec` δ transfers δ atomic percent from one element to another. Allowed δ values depend on the phase: coarse ($\delta \in \{10, 15\}$) during the first half of the budget, and fine ($\delta \in \{2, 5\}$) during the second half. This creates order sensitivity.

UTS surrogate. We train a Gradient Boosting Regressor on 539 measured alloys from the MPEA mechanical properties dataset (Borg et al., 2020). The surrogate predicts ultimate tensile strength (UTS) from the six-element composition vector.

Heavy-element UTS scaling. To create a genuine UTS–density tradeoff, we apply a multiplicative scaling factor to the base UTS prediction:

$$\text{UTS}_{\text{eff}} = \text{UTS}_{\text{base}} \cdot \left[\alpha + (1-\alpha) \cdot \min\left(1,\ \frac{h}{h_{\text{thresh}}}\right)\right], \tag{1}$$

[1] https://norvig.com/ngrams/enable1.txt

where $h = 1.0 \cdot c_{\text{Mo}} + 0.6 \cdot c_{\text{Co}} + 0.3 \cdot c_{\text{Ni}}$, $\alpha = 0.30$, and $h_{\text{thresh}} = 18.0$. Without sufficient heavy elements, UTS is capped well below the target, making it impossible to reach the UTS threshold without including density-increasing elements.

Density and recovery cost. Density is computed as the composition-weighted average of elemental densities. To create path dependence, we also track a recovery cost that accumulates when the composition moves away from a known-good region. If recovery cost exceeds 3.8 at `FINALIZE`, the episode fails even if UTS and density targets are met.

Starting composition and observation. The default composition is Fe:25, Ni:15, Cr:20, Co:10, Mn:20, Mo:10 (at.%). For seed-dependent variation, each element is perturbed by $\pm 8\%$ and renormalized. The agent observes the current composition, a noisy UTS estimate (± 30 MPa Gaussian), exact density, a qualitative path-efficiency indicator, the current phase, and recent edit results.

Success criterion. An episode succeeds if and only if all three conditions hold at `FINALIZE`: $\text{UTS}_{\text{eff}} \geq 2{,}000$ MPa, density ≤ 8.20 g/cm^3, and recovery cost < 3.8.

B.5 GB1 Protein Environment Construction

Biological basis. We base this environment on the GB1 fitness landscape from the FLIP benchmark (Dallago et al., 2021), which provides experimentally measured binding fitness for a large set of GB1 variants.

Editable positions. We select four positions with strong epistatic interactions: V39, D40, G41, and V54. Each position admits five residues, yielding a combinatorial space of $5^4 = 625$ variants. The wild-type sequence is VDGV.

True fitness and the epistasis trap. True fitness is looked up from a precomputed table derived from the FLIP data. Only 44 of 625 variants (7.0%) have fitness ≥ 5.0. The environment is designed so that single-site fitness estimates are systematically misleading about combinatorial fitness: top variants require mutations at position 2, despite position 2 appearing harmful under additive local estimates. This creates the intended local-optimum trap.

Stability mechanic. To create path dependence, each mutation carries a stability delta (Table 4). Position 2 mutations are especially destabilizing. If stability drops below a gate threshold of 0.55, further destabilizing mutations are blocked; if stability falls below 0.35, the episode cannot succeed.

Table 4: Stability deltas per mutation. Negative values indicate destabilization. Position 2 (G41) mutations are the most destabilizing but are required for high combinatorial fitness.

Pos 0 (V39)		**Pos 1 (D40)**		**Pos 2 (G41)**		**Pos 3 (V54)**	
AA	Δs	AA	Δs	AA	Δs	AA	Δs
V	0.00	D	0.00	G	0.00	V	0.00
L	-0.02	A	$+0.02$	A	-0.18	A	$+0.03$
I	-0.03	H	-0.05	C	-0.20	G	$+0.05$
F	-0.08	Y	-0.08	F	-0.22	C	-0.06
W	-0.15	W	-0.10	L	-0.25	F	-0.10

Starting sequence and observation. For seed-dependent variation, we apply one random mutation to the wild-type sequence at reset. The agent observes the current sequence, a noisy additive fitness estimate (± 0.20 Gaussian) together with a warning that it may not reflect true combinatorial fitness, a qualitative stability label, available mutations with local estimates, and recent edit history.

Finalize protocol. The agent may attempt `FINALIZE` at any time provided stability ≥ 0.35. True combinatorial fitness is revealed only upon finalization. Success requires both fitness ≥ 5.0 and stability ≥ 0.35.

B.6 Calibration Procedure and Operating-Point Selection

Each environment was calibrated to ensure that random and myopic baselines achieve low but non-zero success rates, privileged oracle baselines achieve high success rates, and LLM agents occupy an intermediate tier with room for prompt-induced variation.

Word Ladder. The main calibration lever is the shortest-path range $[3, 6]$. Pairs with $k_{\text{opt}} \leq 2$ are too easy, while pairs with $k_{\text{opt}} \geq 7$ yield near-zero success for all non-oracle agents.

Alloy. The key parameters are the UTS target, density target, and heavy-element scaling threshold. These were chosen so that the observation-level greedy heuristic remains above random but far below the privileged balanced oracle.

GB1. We swept the fitness target across $\{4.7, 4.8, 5.0\}$ with four policies (Table 10). At all thresholds, the additive greedy baseline achieves 0%, while the privileged oracle remains substantially above zero. We selected fitness_target $= 5.0$ as the main setting because it most clearly separates additive baselines, LLM agents, and oracle baselines.

B.7 Instance Statistics

Table 5: Instance-level statistics for each environment. The reported quantities summarize search-space size, budget, observation noise, and the relative difficulty of random, myopic, privileged, and LLM policies. The intended calibration regime is visible in the separation between these tiers: the tasks are neither trivial for local heuristics nor degenerate for all agents.

	Word Ladder	**Alloy**	**GB1 Protein**
Instance pool size	409 pairs	continuous	625 variants
Budget (steps)	$5k_{\text{opt}}$ (15–30)	8	6
Avg. branching factor	9.9	∼30/step	∼4/step
Observation noise	none	±30 MPa	±0.20
Random baseline SR	0%	4%	4%
Obs-level heuristic SR	82%	22%	0%
Best privileged oracle SR	100%	93%	100%
LLM zero-shot SR	18%	12%	22%
LLM best-prompt SR	30%	16%	38%

B.8 Filtering and Quality Control

Word Ladder. From 500 candidate pairs, we removed 91 pairs with shortest-path distance outside $[3, 6]$. We verified that all retained pairs are connected in the Hamming-1 graph and that neither start nor target word has zero valid neighbors at the initial state.

Alloy. The composition space is continuous, so there is no fixed instance pool. Instead, we verified that jittered starting compositions satisfy element bounds, that at least one feasible composition exists within 8 steps of each starting point, and that the surrogate model predictions are monotonic in expected directions for major elements.

GB1. The 625 variants are exhaustively enumerated from 4 positions $\times$ 5 residues. We verified that the fitness lookup table contains all 625 entries, that wild-type fitness is normalized to 1.0, and that single-site marginal estimates are computed correctly. No variants were excluded from the evaluation pool.

B.9 Reproducibility Notes

All environments are fully deterministic given a seed. The seed controls instance selection, observation noise, and any stochastic baseline behavior. We used seeds 0–49 for the main experiments and seeds 0–99 for baseline evaluations. The LLM was queried via the Together API using `Qwen/Qwen2.5-7B-Instruct-Turbo` with default temperature. Environment code, fitness lookup tables, and word pair lists will be released upon publication.

C Detailed Main Prompt Results

This section provides the full prompt-intervention table underlying the main comparison in Sections 5.1 and 5.2. The main pattern to verify is that format-only few-shot prompting does not consistently improve performance, while larger gains appear only when the intervention changes the editing policy in an environment-specific way.

Table 6: Detailed main prompt-intervention results under $M0$.

Environment	Zero-shot	Few-shot (format)	Few-shot (strategy)	Scaffold	Self-check (generic)	Self-check (task-specific)
Word Ladder (4-letter)	17	13	20	30	3	13
Alloy editing	20	13	3	13	3	20
GB1 sequence editing	22	20	24	12	26	38

D Detailed History Ablation

This section reports the full history-ablation results summarized in Section 5.4. The key pattern is that recent edit history is not uniformly beneficial: it helps when the task rewards cumulative state tracking, but harms performance when recent trajectories reinforce a poor local path.

Table 7: History-ablation results for selected prompt conditions.

Environment	Prompt	$M0$	$M1$	$M3$	Interpretation
Word Ladder	zero-shot	22	20	10	harmful
Word Ladder	scaffold	30	18	10	strongly harmful
Alloy	zero-shot	12	20	25	helpful
Alloy	self-check	18	8	18	non-monotonic
GB1	zero-shot	25	8	8	strongly harmful
GB1	self-check	35	5	5	catastrophically harmful

E Failure Patterns

This section gives the dominant failure patterns underlying the discussion in Section 5.3. The table is intended to show that the main residual errors are not uniform across environments and are predominantly structural rather than syntactic.

Table 8: Representative dominant failures under the main prompt experiment. The table shows that residual errors are predominantly structural rather than purely syntactic, and that the dominant failure mode differs by environment: budget-unaware continuation in Word Ladder, objective-tradeoff collapse in alloy editing, and local-optimum trapping in GB1. Counts in parentheses indicate the number of episodes in which the listed failure was dominant.

Environment	Prompt condition	Dominant failure
Word Ladder	zero-shot	BUDGET-UNAWARE (19)
Word Ladder	few-shot (format)	BUDGET-UNAWARE (23)
Word Ladder	few-shot (strategy)	BUDGET-UNAWARE (19)
Word Ladder	scaffold	BUDGET-UNAWARE (17)
Word Ladder	self-check (generic)	BUDGET-UNAWARE (18), PREMATURE-FINALIZE (8)
Word Ladder	self-check (task-specific)	BUDGET-UNAWARE (20)
Alloy	zero-shot	OBJECTIVE-TRADEOFF (20)
Alloy	few-shot (format)	OBJECTIVE-TRADEOFF (21)
Alloy	few-shot (strategy)	OBJECTIVE-TRADEOFF (23)
Alloy	scaffold	OBJECTIVE-TRADEOFF (22)
Alloy	self-check (generic)	OBJECTIVE-TRADEOFF (20)
Alloy	self-check (task-specific)	OBJECTIVE-TRADEOFF (18)
GB1	zero-shot	surface: 0, structural: 21; LOCAL-OPTIMUM TRAP (21)
GB1	few-shot (format)	surface: 1, structural: 25; LOCAL-OPTIMUM TRAP (24)
GB1	few-shot (strategy)	surface: 0, structural: 10; LOCAL-OPTIMUM TRAP (10)
GB1	scaffold	surface: 9, structural: 19; LOCAL-OPTIMUM TRAP (17)
GB1	self-check (generic)	surface: 0, structural: 23; LOCAL-OPTIMUM TRAP (22)
GB1	self-check (task-specific)	surface: 0, structural: 22; LOCAL-OPTIMUM TRAP (20)

F Example Prompt Templates

This section includes abbreviated prompt templates for the three environments. They are provided to make the intervention design concrete and to clarify the difference between action-format constraints and policy-shaping content.

F.1 Word Ladder as Lexical Editing

System:

```
You are solving a constructive lexical editing task.
At each step, edit exactly one character of the current word.
Your output must be exactly one action.
Use the format: EDIT <position> <letter>.
Do not explain your answer.
```

User:

```
Target: ruly
Current: many
Used path: many
Editable positions and candidate letters:
0 -> {r, w, z}
1 -> {o, u}
2 -> {l, r}
```

```
3 -> {e, s}
Remaining budget: 8
Your action:
```

F.2 Alloy as Compositional Editing

System:

```
You are solving a constructive alloy-editing task.
At each step, increase one element and decrease another by the same amount.
Use exactly one action in the format:
EDIT <inc_element> <dec_element> <delta>
or FINALIZE.
Do not explain your answer.
```

User:

```
Current composition: Fe 25, Ni 20, Cr 20, Co 15, Mn 10, Mo 10
UTS estimate: 610 MPa (target >= 700)
Density estimate: 8.8 (target <= 9.0)
Trajectory stability: moderate
Reversibility warning: low
Phase: coarse
Allowed delta: {10, 15}
Remaining budget: 6
Your action:
```

F.3 GB1 Biological Sequence Editing

System:

```
You are solving a constructive biological sequence-editing task.
At each step, replace one token in the current sequence.
Use exactly one action in the format:
EDIT <position> <token>
or FINALIZE.
Do not explain your answer.
```

User:

```
Current sequence: V39=V D40=W G41=G V54=V
Estimated fitness: 3.85 (target >= 5.0)
Single-site estimates may be misleading due to epistasis.
Stability: moderate
Remaining budget: 5
Your action:
```

G Alloy Exemplar Robustness

This section reports the alloy exemplar-sensitivity analysis referenced in the main text. It shows that prompt effects in alloy editing depend not only on the existence of a demonstration, but also on the specific trajectory induced by that demonstration.

H GB1 Calibration

This section provides the GB1 operating-point validation used to support the chosen threshold in the main experiments. The table shows that additive local heuristics fail across thresholds, while stronger policies remain above zero, indicating a non-trivial but non-degenerate regime.

Table 9: Alloy few-shot exemplar robustness.

Condition	Success rate (%)
Zero-shot	17
Few-shot format	13
Few-shot strategy A	3
Few-shot strategy B	20

Table 10: GB1 operating-point calibration across fitness thresholds. Additive local heuristics fail across all tested thresholds, whereas the LLM and the privileged noisy oracle remain above zero. This supports the claim that the chosen regime is non-trivial but non-degenerate, and that success cannot be explained by additive local improvement alone.

Policy	ft=4.7	ft=4.8	ft=5.0
Additive greedy	0	0	0
Obs. additive greedy	0	0	0
Zero-shot LLM	30	45	40
Few-shot strategy	20	20	20
Privileged noisy oracle	55	55	55

I Word-Length Ablation

This section reports the lexical operating-point ablation over word length. Its purpose is to justify the choice of the 4-letter setting as the most informative control regime for the main study.

Table 11: Word-length ablation for lexical editing. The 4-letter setting provides the most informative operating regime: 3-letter instances are easier overall and leave less room for intervention effects, whereas 5-letter instances are sparser and harder for all methods. The larger scaffold effect at 4 letters therefore motivates its use as the main lexical setting.

Length	Zero-shot	Scaffold	Scaffold effect
3-letter	23	27	+4
4-letter	13	27	+14
5-letter	10	13	+3