UMO-AGENT: UNCERTAINTY-GUIDED MOLECULAR OPTIMIZATION AGENT FOR DRUG DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular optimization against a protein target is an important step in drug discovery, seeking to improve an initial ligand's binding affinity while maintaining drug-likeness, synthetic accessibility, and structural feasibility. We focus on fragment-based optimization, which enables controlled molecular refinement through modular chemical edits. However, the problem poses several technical challenges, including competing objectives, reward hacking, uncertain evaluation feedback, and combinatorial search. Earlier techniques primarily employ heuristic or evolutionary search, which requires careful operator and objective design, or reinforcement learning, which introduces additional policy-training costs and difficulties. Recent LLM-based agents support molecular optimization through tool use and iterative feedback, motivating mechanisms that limit the propagation of unreliable evaluations through molecular edits and accumulated experience. To fill this gap, we introduce UMO-Agent, a training-free LLM-based agent that integrates predictive uncertainty into the control of iterative fragment-based optimization. The agent combines Monte Carlo dropout estimates from a neural affinity predictor with docking consistency, interaction evidence, and molecular-property assessments. Its central design separates candidate screening from acceptance and links the resulting reliability assessment to memory updates and subsequent fragment search. We evaluate UMO-Agent on the CrossDocked2020 benchmark, and the results show that it achieves an average joint Success Rate of 60% across three independent runs, compared with approximately 35% reported for the agent-based baseline CIDD under its original evaluation protocol.

1 INTRODUCTION

Molecular optimization against a protein target aims to refine an initial lead ligand to improve binding affinity while maintaining favorable drug-likeness, synthetic accessibility, and structural feasibility. This paper focuses on fragment-based optimization, which modifies a molecule by adding, replacing, or removing chemical fragments, providing a modular search space for controlled structural changes (Yang et al., 2021; Chen et al., 2021). This process is important in drug discovery because an initially promising binder rarely satisfies all the requirements for further development. Target-specific refinement allows researchers to explore improvements around an existing ligand while preserving useful molecular features and protein–ligand interactions, helping prioritize candidates before costly synthesis and experimental evaluation (Wei & McCammon, 2024; Tropsha et al., 2024).

However, molecular optimization against a protein target presents several interconnected challenges. First, the problem has competing objectives. Modifications that improve binding can increase molecular size or complexity, compromising drug-likeness and synthetic accessibility. Second, optimization can exploit computational scores. For example, an optimizer may obtain increasingly favorable docking scores through excessive molecular growth without corresponding improvements in physical binding affinity, a form of reward hacking (Yoshizawa et al., 2025). Third, evaluating binding affinity involves a difficult trade-off between computational cost and predictive reliability. Physics-based methods such as free-energy perturbation can provide accurate affinity estimates under suitable conditions, but their computational expense limits their use throughout optimization (Wang et al., 2015). Docking offers a cheaper alternative but relies on approximate scoring functions and limited conformational sampling (Trott & Olson, 2010; Huang et al., 2010). Neural affinity predic-

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

tors can improve predictive accuracy, yet their reliability can deteriorate for unfamiliar molecules or protein environments, making uncertainty particularly consequential during search (Rayka & Naghavi, 2025; Chatterjee et al., 2023). Finally, the combinatorial space of molecular modifications makes finding a globally optimal solution generally impractical, motivating iterative search. Each selected edit changes the starting point for subsequent exploration, so an unreliable apparent improvement can redirect the entire trajectory and become misleading experience that is repeatedly reused.

Existing techniques for molecular optimization include heuristic and evolutionary search, reinforcement learning, and, more recently, LLM-based agents. Heuristic and evolutionary approaches depend on carefully designed search operators and objectives (Jensen, 2019; Fu et al., 2022), while reinforcement learning methods additionally incur policy-training costs and require choices in reward design and exploration (Yang et al., 2021; Chen et al., 2021). Structure-conditioned models such as DecompOpt and MolJO support molecular refinement through learned generative mechanisms (Zhou et al., 2024; Qiu et al., 2024). Recently, there has been growing interest in LLM-based agents that optimize molecules through tool use and feedback without updating the LLM parameters. LIDDiA, CIDD, and MolLingo coordinate optimization tools or guide edits using protein–ligand interactions and fragment representations (Averly et al., 2025; Gao et al., 2026; Nguyen & Ji, 2026a). SEISMO uses trajectory history and structured feedback, including prediction confidence in protein-target experiments (Krüger et al., 2026), while SpecOpt guides edits through target–off-target contact differences (Nguyen & Ji, 2026b). The closest related work is PROBE (Yang et al., 2026), which performs controlled fragment edits to measure pocket-specific responses, summarizes these observations in an EditManual, and guides affinity–druggability optimization. However, PROBE does not explicitly quantify predictive uncertainty in those edit responses to determine how strongly they should guide subsequent search. Uncertainty-aware molecular optimization has also been studied through reliability-based reward design (Yoshizawa et al., 2025; Chen et al., 2026), probabilistic candidate selection (Nguyen & Grover, 2025; Ranković et al., 2026), and exploration-driven experience replay using Monte Carlo dropout (Yang et al., 2021). These developments motivate a more specific question for molecular agents: how should uncertainty in an apparent improvement influence whether an edit is retained, whether its outcome reinforces target-local experience, and how subsequent edits are explored?

To fill this gap, we introduce UMO-Agent, a training-free LLM-based agent that integrates affinity-prediction uncertainty into the control of iterative molecular optimization. Given a protein–ligand complex, the agent uses pocket information and target-local experience to guide stochastic search over a fixed library of fragment edits. Its central design separates the screening of promising candidates from the evidence required to retain an improvement. Monte Carlo dropout estimates (Gal & Ghahramani, 2016) from a neural affinity model are combined with fresh docking evaluations, interaction evidence, and molecular-property assessments to inform this decision. The outcomes also update target-local memory: improvements supported by re-evaluation reinforce productive edit directions, while evaluation failures and measured risk inform subsequent search. Our contribution lies in coupling candidate acceptance, experience updates, and future edit selection through a shared assessment of evaluation reliability. We evaluate UMO-Agent on the CrossDocked2020 benchmark alongside representative molecular optimization methods. Across three independent runs, our approach achieves an average joint Success Rate of 60%, compared with approximately 35% reported for the agent-based baseline CIDD under its original evaluation protocol (Gao et al., 2026). Component ablations support the effectiveness of the combined framework, with lower joint success when memory, adaptive proposal mechanisms, or the verification are removed.

2 Problem Definition

Given a target protein pocket p, an initial bound ligand x_0, a fragment library $\mathcal{F}$, and a computational evaluation budget, the task is to return an optimized molecule $x^\star$ obtained through successive fragment edits to x_0. The objectives are to improve predicted target-binding affinity while achieving favorable drug-likeness and synthetic accessibility, measured by QED and normalized SA, respectively. Optimization is subject to the following constraints: admissible fragment edits comprise addition, replacement, and deletion; each resulting molecule must satisfy structural-validity requirements, and each edit must respect specified spatial and molecular-growth limits. Binding affinity is assessed computationally using AutoDock Vina and a pretrained neural affinity predictor, with

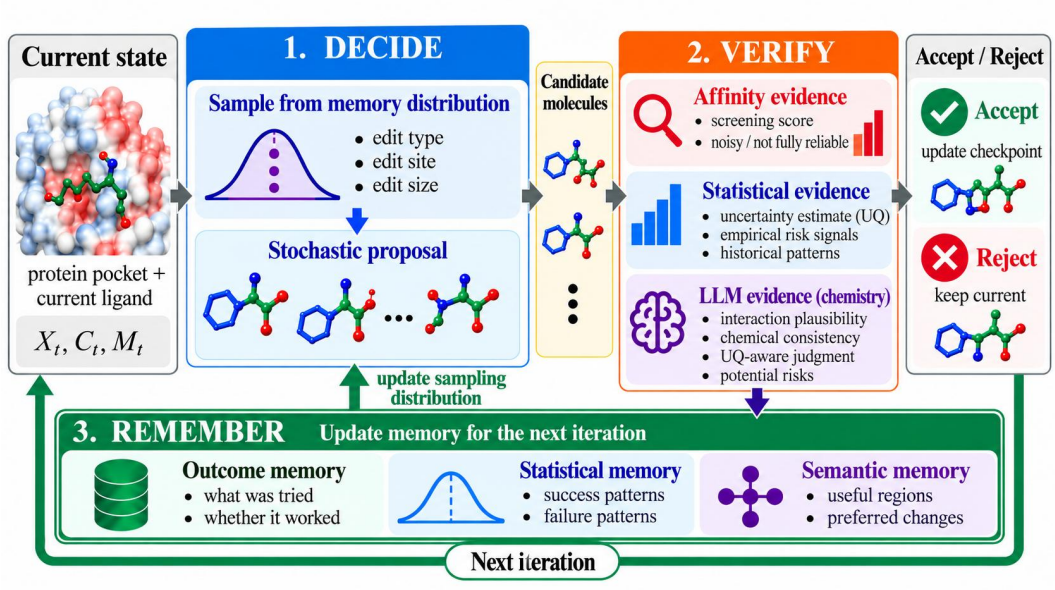


Figure 1: **Overview of UMO-Agent.** The agent samples multiple fragment edits from a memory-guided distribution, screens the resulting candidates, and applies full docking verification and reliability assessment to one selected challenger. Acceptance updates the retained molecule; evaluation outcomes update target-local memory and subsequent sampling.

evaluation calls restricted by the available budget. These evaluations provide approximate guidance for optimization.

3 Proposed Method

3.1 Overview

UMO-Agent couples fragment-based molecular search with an assessment of evaluation reliability. Its central idea is to use the evidence supporting an apparent improvement to inform both molecular acceptance and the experience that guides subsequent search. At iteration t, the agent maintains a retained molecule c_t, initialized as $c_0 = x_0$, and target-local memory M_t. Pocket information and accumulated experience guide a distribution over fragment edits. The agent samples and screens multiple candidates, selects one challenger for full verification, and decides whether it should replace c_t. The observed outcome then updates memory and the next sampling distribution, completing the loop in Fig. 1. Model parameters remain fixed, and the final retained molecule is returned when the evaluation budget or stopping criterion is reached.

The loop has two coupled effects: acceptance determines the starting molecule for the next iteration, while memory determines which edit directions receive attention. A rejected proposal can therefore be informative even though it does not change the retained molecule. Separating these effects allows the agent to explore candidate modifications while retaining the evaluation evidence needed to assess and reuse their outcomes.

3.2 Memory-Guided Candidate Sampling

A high-level edit specifies its operation, editable region, magnitude, search scope, and fragment profile. For each factor $d \in \mathcal{D}$, let $\mathcal{K}_d$ denote its admissible values. Molecular editing tools instantiate a concrete transformation from the compatible fragment library after these factors are sampled. The LLM Pocket Planner receives deterministic pocket–ligand descriptors, current objective deficits and property margins, and semantic memory. It returns bounded categorical preferences over edit factors. The descriptors include available pocket space, polarity, nearby residues, and existing protein–ligand interactions. For example, the planner may favor a small addition near an accessible region

while preserving an existing interaction. The editing tools then sample the attachment site and compatible fragment within the allowed categories. Operations include ADD, REPLACE, and DELETE, while search scope distinguishes LOCAL from BROAD exploration. Thus, each sampled decision specifies a region of the edit space without requiring the LLM to emit an exact molecular structure.

The proposal score for factor value $k \in \mathcal{K}_d$ is

$$S_t^{(d)}(k) = O_t^{(d)}(k) - C_t^{(d)}(k) + \beta_U U_t^{(d)}(k) - \lambda_R(r_t)\rho_t^{(d)}(k) + \log b_{\mathrm{struct},t}^{(d)}(k) + \lambda_{\mathrm{LLM},t} \log b_{\mathrm{LLM},t}^{(d)}(k). \tag{1}$$

Here, $O_t^{(d)}(k)$ estimates verified opportunity, and $C_t^{(d)}(k)$ aggregates the QED, SA, and molecular-growth costs detailed in Appendix A.5. The exploration bonus $U_t^{(d)}(k) = 1/\sqrt{n_t^{(d)}(k) + 1}$ uses the previous trial count $n_t^{(d)}(k)$ to favor less frequently tried decisions. Verification failures and measured risk events contribute to $\rho_t^{(d)}(k)$. The planning reliability state r_t uses information available before the current proposal; $\lambda_R(r_t)$ is zero in GREEN states and becomes active in YELLOW or RED states. The final terms encode deterministic structural preferences and bounded LLM preferences, respectively. The LLM weight decreases as verified experience accumulates.

The score separates how often an edit direction has produced useful outcomes from its property cost and reliability history. The count-based exploration bonus rewards under-sampled factors; it is distinct from the neural predictive uncertainty assessed after candidate selection. Structural and LLM preferences enter as positive multipliers in log space, so a neutral multiplier of one contributes zero. Early iterations can benefit from pocket-informed preferences, while later iterations give greater influence to evidence collected on the current target.

Scores define an exploration-mixed Boltzmann distribution,

$$\pi_t^{(d)}(k) = (1-\epsilon)\frac{\exp(S_t^{(d)}(k)/T)}{\sum_{j\in\mathcal{K}_d}\exp(S_t^{(d)}(j)/T)} + \frac{\epsilon}{|\mathcal{K}_d|}. \tag{2}$$

Temperature T controls concentration, and the uniform component preserves exploration. Let $\widetilde{\pi}_t^{(d)}$ denote the probabilities after the additional capping and renormalization step described in Appendix A.5. The high-level edit distribution factorizes as $\pi_t(a) = \prod_{d\in\mathcal{D}} \widetilde{\pi}_t^{(d)}(a^{(d)})$. The softmax favors high-scoring factors without making selection deterministic; the uniform mixture keeps alternatives reachable even after several favorable observations concentrate the learned preferences.

The agent samples N_t edits and screens the structurally valid candidates using inexpensive docking and molecular-property evaluations. The default initial pool contains 40 edits, providing several alternatives from the same retained molecule before a challenger is selected. Ranking prioritizes affinity improvement, property repair, or balanced improvement according to the current objective deficits. For example, when affinity already meets its target but QED or SA remains deficient, ranking favors property repair while preserving the available affinity margin. When several objectives are deficient, a normalized multi-objective score balances their improvements. The strongest admissible candidate becomes challenger $x_t^\star$, with associated edit $a_t^\star$, for full verification. If the initial pool contains no productive candidate, the pool is expanded before declaring local convergence. Selection rules and numerical settings are in Appendix A.4.

3.3 Uncertainty-Informed Verification and Acceptance

During screening, candidate poses are constructed by constrained embedding around the retained ligand's pose and evaluated through local optimization. This makes comparison across the candidate pool inexpensive, but the score depends on the inherited pose. The selected challenger and retained molecule subsequently undergo fresh, uncached full docking under the same protocol. Broader pose exploration can reveal that a screening gain does not persist, or that a promising candidate adopts a different binding pose. Their average docking difference is

$$g_t^{\mathrm{cert}} = \frac{1}{K_{\mathrm{dock}}}\sum_{j=1}^{K_{\mathrm{dock}}}\left[V_j(c_t;p) - V_j(x_t^\star;p)\right]. \tag{3}$$

Here, K_{dock} is the number of docking repetitions and $V_j(x;p)$ is the score for molecule x against pocket p in repetition j. Lower Vina scores indicate more favorable predicted binding, so positive g_t^{cert} indicates improvement under the verification protocol. This comparison evaluates whether screening gains persist under broader pose exploration. Both stages use the same approximate scoring function, so additional docking does not establish an improvement in true binding affinity.

The agent also estimates neural affinity-prediction uncertainty through Monte Carlo dropout (Gal & Ghahramani, 2016). Repeated stochastic forward passes provide an auxiliary predictive-uncertainty signal u_t. The molecular input remains fixed across passes, allowing variation in predictions to inform the assessment of the neural evaluator. The reported configuration uses five passes. This signal accompanies the predicted affinity so that a favorable neural prediction can be considered together with the variability supporting it. The evidence vector $\mathbf{e}_t$ combines this signal with docking gain, screening–verification reversal, pose consistency, unusually large local gain, evaluator disagreement, interaction changes, molecular properties, and recent verification history.

An LLM critic returns separate concern levels for measurement-supported evidence, C_t^{evid}, and broader chemical considerations, C_t^{chem}, each in $\{\mathrm{LOW}, \mathrm{MODERATE}, \mathrm{HIGH}\}$. A deterministic controller assigns

$$R_t = \mathrm{Ctrl}(\mathbf{e}_t, C_t^{\mathrm{evid}}, C_t^{\mathrm{chem}}) \in \{\mathrm{GREEN}, \mathrm{YELLOW}, \mathrm{RED}\}. \tag{4}$$

Unlike the prior planning state r_t, R_t assesses the current challenger. RED requires a severe measured anomaly or a high evidence concern corroborated by candidate-specific measurements. Elevated neural uncertainty, evaluator disagreement, or chemical concern alone cannot trigger RED. The critic's outputs are advisory; measured evidence and controller rules determine acceptance (Appendix A.7).

For example, loss of a favorable screening result after full docking provides candidate-specific evidence of evaluation inconsistency. A general concern that a large or polar substituent could behave poorly is instead a chemical consideration unless supported by measurements. Keeping the two concern channels separate prevents a plausible explanation from being counted as an observed failure. A non-RED assessment still requires the candidate to meet the feasibility and improvement conditions below.

Let $I_t(x_t^\star, c_t) \in \{0,1\}$ indicate whether the verified candidate meets the improvement requirement for the current optimization mode. In affinity optimization, this requires $g_t^{\mathrm{cert}} > \delta_{\mathrm{cert}}$. Property repair instead requires improvement in a deficient property while preserving the required affinity margin. The acceptance decision is

$$A_t = \mathbb{I}[\mathrm{Feasible}(x_t^\star)]\, I_t(x_t^\star, c_t)\, \mathbb{I}[R_t \neq \mathrm{RED}]. \tag{5}$$

If $A_t = 1$, then $c_{t+1} = x_t^\star$; otherwise, $c_{t+1} = c_t$. Rejection preserves the retained molecule while the evaluation outcome remains available to guide subsequent search.

3.4 Memory Update and Sampling Adaptation

Memory maintains outcome records, statistical summaries, and semantic lessons. Each verified edit supplies its high-level factors, molecular-property changes, screening and verification results, reliability assessment, and acceptance decision. The records distinguish apparent screening gains from improvements supported by additional evaluation.

Statistical memory tracks trial counts, verified successes, and verified gains for edit-factor values. The opportunity estimate combines the probability of verified success, using a $\mathrm{Beta}(1,1)$ prior, with the expected positive verified gain. Verification failures and measured risk events contribute to the risk statistic. Only verification-supported outcomes provide positive opportunity evidence; a favorable screening score alone does not. Success probability summarizes how consistently an edit direction produces verification-supported improvements, while the gain estimate summarizes their magnitude. The prior supplies an initial estimate when few observations are available. These quantities and the risk statistics feed directly into $O_t^{(d)}(k)$ and $\rho_t^{(d)}(k)$ in Eq. 1, closing the connection between evaluated outcomes and subsequent sampling. Appendix A.6 describes the statistical memory.

For illustration, consider a sampled addition that improves the local screening score but fails the subsequent docking comparison. The retained molecule remains unchanged, and the edit contributes

a verification failure rather than positive opportunity evidence. Repeated measured failures can increase the corresponding risk statistic, which affects later sampling when the risk penalty is active. The exploration component nevertheless leaves alternative trials possible.

The Experience Summarizer converts structured statistics into lessons about productive regions, interactions to preserve, and unsuccessful or risky edit patterns. Each lesson retains numerical support from the underlying records and is supplied to later Pocket Planner calls. Semantic summaries therefore compress measured experience without creating new verified successes. Memory resets for each target, and all updates use outcomes observed before the next proposal. The resulting loop adapts both the retained molecule and the allocation of search effort through accumulated evaluation evidence.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate on the CrossDocked2020 benchmark Francoeur et al. (2020) using the 100-target TargetDiff test split Guan et al. (2023a). Each trajectory starts from the bound reference target, which is also reported as the Reference row. Because optimization performance is strongly initialization-dependent, we focus direct comparisons on methods with similar optimization setups: RGA Fu et al. (2022), DecompOpt Zhou et al. (2024), MolJO Qiu et al. (2024), and CIDD Gao et al. (2026). CIDD also reports a hit-to-lead setting initialized from ground-truth ligands in the CrossDocked test set, providing a closely matched starting point to our reference-ligand optimization setup, although the downstream optimization protocol differs. Representative structure-based generative models Peng et al. (2022); Guan et al. (2023a;b); Qu et al. (2024) are included only for broader benchmark context. Baseline values are taken from the corresponding publications.

Following prior benchmarks (Guan et al., 2023b; Qu et al., 2024; Qiu et al., 2024), we report the full-docking Vina score in kcal/mol (lower is better), QED, normalized synthetic accessibility (SA; higher is better), and joint Success Rate (SR). SR is the percentage of delivered molecules simultaneously satisfying Vina < -8.18, QED > 0.25, and SA > 0.59. Our method delivers one molecule per target per run. To characterize the trade-off between docking-score improvement and molecular growth, the component analysis additionally reports heavy-atom change relative to the initial ligand (ΔHA), ligand efficiency ($\mathrm{LE} = \mathrm{Vina}/N_{\mathrm{HA}}$; lower is better under this sign convention), and average trajectory length in optimization steps.

AutoDock Vina Trott & Olson (2010) provides both local candidate screening and fresh full-docking verification; these are different evaluation procedures using the same scoring function, rather than independent models of binding affinity. All molecular models, scientific evaluators, and LLM components remain frozen, with LLM modules using claude-4.8-opus. We repeat the full method's evaluation with three random seeds and report mean and standard deviation in Table 1. Published baselines are point estimates under their original protocols. Appendix A provides implementation and metric details.

4.2 OVERALL PERFORMANCE

Table 1 shows that UMO-Agent improves the mean Vina score of the reference ligands from -7.450 to -8.636, while increasing joint SR from 25% to 60%. Mean QED increases from 0.480 to 0.526, whereas mean normalized SA decreases from 0.730 to 0.683. These changes expose the trade-off involved in optimizing the docking objective: improved docking scores do not imply improvement in every molecular property. The joint SR measures how often all three benchmark thresholds are satisfied; mean property values alone cannot establish this.

UMO-Agent has the highest reported joint SR among the methods in the table, including CIDD's 34.59%, although several baselines achieve stronger individual metrics. For example, DecompOpt and MolJO report more favorable mean Vina scores, and MolJO reports higher SA. Notably, CIDD additionally reports a hit-to-lead extension initialized from ground-truth ligands in the CrossDocked test set, matching our reference-ligand starting point more closely than its standard generation setting. Under this reported hit-to-lead setting, CIDD achieves an SR of 37.04%, whereas UMO-Agent reaches $60\% \pm 3\%$, an absolute improvement of 22.96 percentage points. Because baseline results

Table 1: **Results on the CrossDocked2020 100-pocket test benchmark.** Reference denotes the original bound test ligands. Baseline values are reproduced from the corresponding publications and follow their original generation or optimization protocols. Our results are reported as mean ± standard deviation over three independent runs.

Method	Vina ↓	QED ↑	SA ↑	SR (%) ↑
Reference	-7.450	0.480	0.730	25.00
Representative Generative Models				
Pocket2Mol	-7.150	0.560	0.740	24.40
TargetDiff	-7.800	0.480	0.580	10.50
DecompDiff	-8.390	0.450	0.610	24.50
MolCRAFT	**-9.250**	0.460	0.620	36.10
Optimization / Goal-Directed Methods				
RGA + Vina	-8.010	0.570	0.710	46.20
DecompOpt	-8.980	0.480	0.650	52.50
MolJO	-9.050	0.560	**0.780**	51.30
CIDD	-8.496	**0.576**	0.735	34.59
CIDD (hit-to-lead)	-8.220	0.528	0.752	37.04
Ours	-8.636 ± 0.066	0.526 ± 0.035	0.683 ± 0.023	$\mathbf{60.00 \pm 3.00}$

Table 2: **Component variants and a shared-action-space search baseline on CrossDocked2020.** The three component variants retain full-docking verification. Greedy Search shares the objective and edit action space but samples uniformly and accepts on the screening score alone. All reported Vina values are obtained by full docking under the same evaluation protocol. ΔHA denotes the change in heavy-atom count, LE denotes ligand efficiency, and Steps denotes the average number of optimization steps before termination. Results are reported as mean ± standard deviation over independent runs.

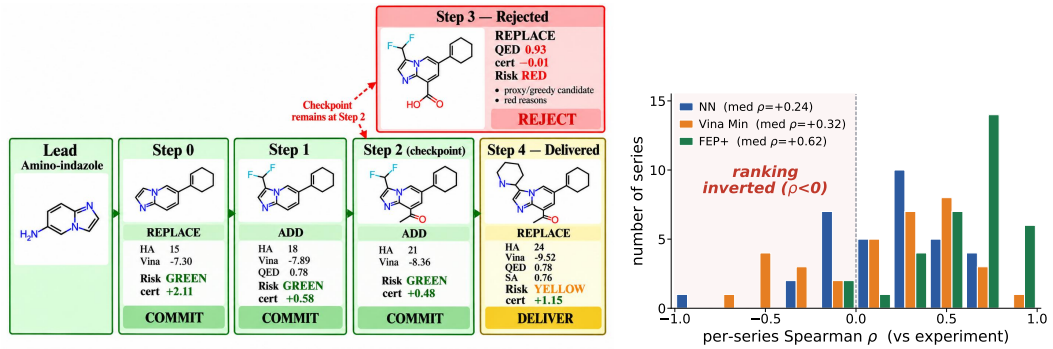
Method	Vina ↓	QED ↑	SA ↑	SR (%) ↑	ΔHA	LE ↓	Steps ↓
Greedy Search	-8.191 ± 0.077	$\mathbf{0.632 \pm 0.024}$	0.651 ± 0.017	36 ± 3	$+6.6 \pm 1.2$	-0.241 ± 0.022	8.0 ± 0.3
Full	-8.636 ± 0.066	0.526 ± 0.035	$\mathbf{0.683 \pm 0.023}$	$\mathbf{60 \pm 3}$	$\mathbf{+2.2 \pm 1.2}$	$\mathbf{-0.309 \pm 0.028}$	$\mathbf{3.2 \pm 0.5}$
w/o Memory	-8.471 ± 0.079	0.443 ± 0.040	0.662 ± 0.029	51 ± 4	$+2.6 \pm 1.6$	-0.306 ± 0.039	4.8 ± 0.3
w/o Decision	-8.594 ± 0.065	0.386 ± 0.023	0.646 ± 0.022	40 ± 2	$+6.4 \pm 1.2$	-0.276 ± 0.035	4.8 ± 0.4
w/o Verification	$\mathbf{-8.787 \pm 0.098}$	0.457 ± 0.036	0.621 ± 0.028	43 ± 3	$+6.5 \pm 1.7$	-0.295 ± 0.014	4.9 ± 0.2
w/o MC-Dropout	-8.650 ± 0.065	0.530 ± 0.028	0.700 ± 0.011	50 ± 4	$+6.6 \pm 1.0$	-0.289 ± 0.022	3.4 ± 0.4

come from different generation or optimization protocols, with differences in sampling budgets, molecular sizes, evaluator use, and other implementation details, these comparisons provide benchmark context rather than a controlled ranking of optimization ability. The reported $60\% \pm 3\%$ summarizes variation across three runs of UMO-Agent; it does not establish statistical significance against the published baseline estimates.

The component analysis below examines the trade-off more directly within our framework: removing the verification improves the mean docking score but reduces joint SR and increases molecular growth. This observation supports evaluating molecular optimization through joint criteria rather than the docking score alone.

4.3 Component Analysis

We compare the full framework with variants that remove memory-guided proposal terms (w/o Memory), adaptive proposal mechanisms (w/o Decision), or the evidence-based RED gate (w/o Verification). All three variants retain full-docking verification. The decision variant disables both the planner and memory-guided proposal bias, so it assesses their combined role in proposal allocation rather than the LLM alone. We also compare against **Greedy Search**, which shares the fragment library, edit operations, candidate-pool size, and multi-objective ranking, but samples uniformly and accepts improvements in the screening score without full-docking verification, risk gating, or memory. Appendix A.10 gives the variant definitions.



(a) **(b)**

Figure 2: **Reliability-aware optimization in practice and across targets.** (a) A representative trajectory shows how the agent commits certified improvements, rejects an attractive but unsupported candidate, and preserves the previous checkpoint. (b) Across 34 congeneric series, evaluator rank correlation with experimental affinity varies substantially across targets, motivating explicit certification during sequential optimization.

Removing the verification yields a more favorable Vina score (-8.78 versus -8.64), but joint SR falls from 60% to 43% and mean growth increases from 2.2 to 6.5 heavy atoms. QED and SA also decrease. Thus, the additional docking-score improvement comes with weaker performance on the joint objective. This comparison supports the role of the verification in balancing the measured objectives; full-docking verification alone does not establish physical binding-affinity improvement. To further assess the role of uncertainty quantification in verification, we additionally remove only the MC-dropout predictive-uncertainty signal while retaining all other verification and risk-control components; this reduces joint SR from 60% to 50% and increases molecular growth from $+2.2$ to $+6.6$ heavy atoms.

Removing memory reduces joint SR to 51%, while removing adaptive proposal mechanisms reduces it to 40% and increases growth to 6.4 heavy atoms. These results support the use of target-local experience and adaptive proposal allocation in the combined framework. Greedy Search achieves 36% joint SR and adds 6.6 heavy atoms on average, with less favorable ligand efficiency than the full method (-0.241 versus -0.309). Because Greedy changes sampling, memory, and acceptance together, its performance cannot be attributed to the acceptance rule alone.

The full method uses an average of 3.2 optimization steps, compared with 4.8–4.9 for the component variants and 8.0 for Greedy Search. These trajectory lengths describe search behavior, not wall-clock or evaluator-call efficiency: variants use different evaluation procedures, and the candidate pool can expand during search. Overall, the reported point estimates support the broader components but do not isolate the contribution of predictive uncertainty or establish the statistical significance of individual ablation differences.

UMO-Agent is also robust to the choice of LLM backend, with joint SR remaining between 60% and 62% across Claude-4.8 Opus, GPT-5.4, and GPT-OSS (Appendix B.5).

4.4 Evaluator Reliability at the Optimization Scale

The premise of this work is that molecular optimization relies on feedback that can be unreliable at the scale of local edits. We test this directly using the congeneric series curated by Ross et al. (2023), which contain shared scaffolds with small substituent changes and measured binding affinities. We compare AutoDock Vina, reported relative free-energy calculations, and a pretrained neural affinity predictor (Zhang et al., 2026). Figure 2b summarizes rank correlation with experiment across 34 series. Free energy performs best overall, with median Spearman correlation 0.62, compared with 0.32 for docking and 0.24 for the neural predictor, and outperforms docking on $27/34$ series ($p < 10^{-3}$, median paired difference $+0.29$).

More importantly, the inexpensive evaluators exhibit target-specific failures that can actively misdirect optimization. Both docking and the neural predictor have negative rank correlation on nine series, while docking falls below -0.3 on seven, indicating systematic inversion rather than merely

weak signal. Such errors can be amplified by search because optimization preferentially selects candidates favored by the evaluator, consistent with the ligand-growth behavior observed in Section 4.3. These failures are not uniform across evaluators: for example, on cr_bace2 docking reaches -0.54 while the neural predictor reaches $+0.57$. Evaluator disagreement therefore provides useful evidence even when neither model is individually reliable.

Higher-fidelity evaluation reduces but does not eliminate the problem: free energy remains below 0.45 correlation on ten of the 34 series and is near chance on two. Thus, our framework does not assume that any evaluator is uniformly reliable or attempt to correct its predictions. Instead, it conditions commitment on the evidence supporting each candidate, allowing uncertainty, evaluator disagreement, and verification outcomes to influence whether an apparent improvement is retained.

4.5 Case Study

Figure 2a illustrates this behavior on a weak amino-indazole fragment (HA 15, Vina -7.30). The agent first replaces the scaffold with an imidazopyridine core (Step 0, cert $+2.11$), then adds a difluoromethyl group (Step 1, cert $+0.58$) and an acetyl group (Step 2, cert $+0.48$), all committed under GREEN risk, reaching Vina -8.36 at this checkpoint. At Step 3, the agent attempts replacing the acetyl group with a carboxylic acid, raising QED to 0.93, the highest value in the entire trajectory. However, the certificate collapses to -0.01 and the risk classifier flags RED as a proxy/QED-greedy candidate, so the edit is rejected and the checkpoint remains at Step 2. This step exposes a failure mode of QED itself: a target-agnostic descriptor score can reward a chemically unremarkable substitution (here, replacing a neutral ketone with an ionizable carboxylate) even when it is incompatible with the specific binding pocket, confirming that QED must be corroborated by a target-grounded certificate rather than used as a stopping criterion on its own. Having just observed this rejection, the classifier becomes more cautious toward large edits, but with no concrete evidence of a specific liability it only flags the next large edit (Step 4, difluoromethyl $\rightarrow$ tetrahydropyridine) YELLOW rather than REJECT. The edit is delivered despite its positive certificate ($+1.15$), and the accumulated risk then triggers early termination of the search.

5 Limitations

Our framework uses uncertainty and consistency signals to reduce the propagation of unreliable evaluator feedback, but it cannot guarantee that the underlying affinity estimates are correct, particularly when multiple evaluators fail consistently on the same target. Higher-fidelity physical or experimental measurements could provide stronger verification, but are currently too expensive for routine use throughout optimization. In addition, the LLM components are integrated into the overall control flow, and their individual contribution is not fully isolated by the current experiments.

6 Conclusion and Future Work

We presented UMO-Agent, a training-free framework for fragment-based molecular optimization that combines predictive uncertainty with docking, interaction, and molecular-property evidence to guide acceptance, memory updates, and subsequent search. On CrossDocked2020, the framework improves joint benchmark success relative to the initial ligands, and component comparisons support the combined design. The results also demonstrate why a more favorable docking score alone is insufficient to assess optimization quality. Future work will examine higher-fidelity physical or experimental validation, the contribution of individual reliability signals, and transfer of search experience across protein targets.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code generation and manuscript writing. Parts of the experimental and analysis code were generated with the assistance of large language models (LLMs) under author supervision. The generated code was reviewed, modified when necessary, and tested by the authors to verify its correctness and consistency with the intended methodology. Generative AI tools were also used to assist with drafting and polishing portions of the manuscript. The authors subsequently reviewed, revised, and edited all AI-assisted text to ensure technical accuracy, clarity, and consistency with the actual methods and experimental results. We take full responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI. LLMs were also used as components of the proposed research methodology. As part of our method, an LLM is incorporated into the agentic optimization framework to support high-level reasoning and decision making. Its role, inputs, and outputs are described explicitly in the main paper and appendix.

REFERENCES

Reza Averly, Frazier N Baker, Ian A Watson, and Xia Ning. Liddia: Language-based intelligent drug discovery agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12015–12039, 2025.

Ayan Chatterjee, Robin Walters, Zohair Shafi, Omair Shafi Ahmed, Michael Sebek, Deisy Gysi, Rose Yu, Tina Eliassi-Rad, Albert-László Barabási, and Giulia Menichetti. Improving the generalizability of protein-ligand binding predictions with AI-Bind. *Nature Communications*, 14:1989, 2023. doi: 10.1038/s41467-023-37572-z.

Benson Chen, Xiang Fu, Regina Barzilay, and Tommi S. Jaakkola. Fragment-based sequential translation for molecular optimization. In *NeurIPS 2021 Workshop on AI for Science: Mind the Gaps*, 2021.

Lianghong Chen, Dongkyu Kim, Mike Domaratzki, and Pingzhao Hu. Uncertainty-aware multi-objective reinforcement learning-guided diffusion models for 3d de novo molecular design. *Advances in Neural Information Processing Systems*, 38:124708–124748, 2026.

Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. 3d convolutional neural networks and a crossdocked dataset for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200, 2020.

Tianfan Fu, Wenhao Gao, Connor Coley, and Jimeng Sun. Reinforced genetic algorithm for structure-based drug design. *Advances in Neural Information Processing Systems*, 35:12325–12338, 2022.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Bowen Gao, Yanwen Huang, Yiqiao Liu, Wenxuan Xie, Bowei He, Haichuan Tan, Wei-Ying Ma, Ya-Qin Zhang, and Yanyan Lan. Cidd: Collaborative intelligence for structure-based drug design empowered by llms. *Advances in Neural Information Processing Systems*, 38:13651–13699, 2026.

Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. *arXiv preprint arXiv:2303.03543*, 2023a.

Jiaqi Guan, Xiangxin Zhou, Yuwei Yang, Yu Bao, Jian Peng, Jianzhu Ma, Qiang Liu, Liang Wang, and Quanquan Gu. Decompdiff: Diffusion models with decomposed priors for structure-based drug design. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11827–11846, 2023b.

Sheng-You Huang, Sam Z Grinter, and Xiaoqin Zou. Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions. *Physical Chemistry Chemical Physics*, 12(40):12899–12908, 2010.

Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

Fabian P. Krüger, Andrea Hunklinger, Adrian Wolny, Tim J. Adler, Igor Tetko, and Santiago David Villalba. SEISMO: Explanation-aware, trajectory-conditioned LLM agents for sample-efficient molecular optimisation. *arXiv preprint arXiv:2602.00663*, 2026.

Thao Nguyen and Heng Ji. Mollingo: Molecule-native representations for llm-powered scientific agents. *arXiv preprint arXiv:2605.27853*, 2026a.

Thao Nguyen and Heng Ji. SpecOpt: Contact-diff reasoning for agentic molecule optimization toward binding specificity. *arXiv preprint arXiv:2609.21165*, 2026b.

Tung Nguyen and Aditya Grover. Lico: Large language models for in-context molecular optimization. In *International Conference on Learning Representations*, volume 2025, pp. 53576–53597, 2025.

Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International conference on machine learning*, pp. 17644–17655. PMLR, 2022.

Keyue Qiu, Yuxuan Song, Jie Yu, Hongbo Ma, Ziyao Cao, Zhilong Zhang, Yushuai Wu, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. Empower structure-based molecule optimization with gradient guided bayesian flow networks. *arXiv preprint arXiv:2411.13280*, 2024.

Yanru Qu, Keyue Qiu, Yuxuan Song, Jingjing Gong, Jiawei Han, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. Molcraft: structure-based drug design in continuous parameter space. *arXiv preprint arXiv:2404.12141*, 2024.

Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery. *Nature Machine Intelligence*, 8: 1466–1477, 2026. doi: 10.1038/s42256-026-01283-z.

Milad Rayka and S. Shahab Naghavi. Uncertainty quantification enables reliable deep learning for protein–ligand binding affinity prediction. *Scientific Reports*, 15:43156, 2025. doi: 10.1038/s41598-025-27167-7.

Gregory A Ross, Chao Lu, Guido Scarabelli, Steven K Albanese, Evelyne Houang, Robert Abel, Edward D Harder, and Lingle Wang. The maximal and current accuracy of rigorous protein-ligand binding free energy calculations. *Communications Chemistry*, 6(1):222, 2023.

Alexander Tropsha, Olexandr Isayev, Alexandre Varnek, Gisbert Schneider, and Artem Cherkasov. Integrating qsar modelling and deep learning in drug discovery: the emergence of deep qsar. *Nature Reviews Drug Discovery*, 23(2):141–155, 2024.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Lingle Wang, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K Dahlgren, Jeremy Greenwood, et al. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *Journal of the American Chemical Society*, 137 (7):2695–2703, 2015.

Haixin Wei and J Andrew McCammon. Structure and dynamics in drug discovery. *npj Drug Discovery*, 1(1):1, 2024.

Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. Hit and lead discovery with explorative RL and fragment-based molecule generation. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Zaifei Yang, Weiyu Chen, Yaqing Wang, and James Kwok. Probe before you edit: Probing-guided molecular optimization for llm agents in structure-based drug design. *arXiv preprint arXiv:2606.00555*, 2026.

Tatsuya Yoshizawa, Shoichi Ishida, Tomohiro Sato, Masateru Ohta, Teruki Honma, and Kei Terayama. A data-driven generative strategy to avoid reward hacking in multi-objective molecular design. *Nature communications*, 16(1):2409, 2025.

Yupu Zhang, Zelin Xu, Tingsong Xiao, Gustavo Seabra, Yanjun Li, Chenglong Li, and Zhe Jiang. Decoydb: A dataset for graph contrastive learning in protein-ligand binding affinity prediction. *Advances in Neural Information Processing Systems*, 38, 2026.

Xiangxin Zhou, Xiwei Cheng, Yuwei Yang, Yu Bao, Liang Wang, and Quanquan Gu. Decompopt: Controllable and decomposed diffusion models for structure-based molecular optimization. In *International Conference on Learning Representations*, volume 2024, pp. 5890–5912, 2024.

A Experimental Details

A.1 Task and Dataset

We study structure-based molecular optimization rather than de novo molecular generation. Given a protein pocket p and its bound reference ligand x_0, the optimizer iteratively modifies the ligand through local fragment-level edits with the goal of improving predicted binding affinity while maintaining drug-likeness and synthetic accessibility. Every optimization trajectory therefore begins from an experimentally associated reference ligand rather than from an empty pocket or a randomly generated molecule.

We evaluate on the CrossDocked2020 test split following the 100-pocket split commonly used in structure-based molecular generation. For each target, the reference ligand serves as the initial optimization state and the corresponding 10 Å-truncated protein pocket is used as the receptor. The initial ligands span a broad range of difficulty, with docking scores ranging approximately from -4 to -9.7, QED values from 0.1 to 0.85, and molecular sizes from 8 to 34 heavy atoms. All 100 test ligands contain at least one valid attachment site under our fragment-editing rules.

A.2 Docking and Molecular Evaluation

We use AutoDock Vina as the primary structure-based affinity evaluator. Receptors are prepared once for each target using Meeko after removing non-protein HETATM records. The docking box is centered at the centroid of the reference ligand and extends the ligand bounding box by 12 Å along each axis. Unless otherwise specified, Vina uses exhaustiveness 8, returns up to 5 poses, and uses one CPU thread per docking worker.

To separate inexpensive candidate discovery from more expensive molecular commitment, we employ a two-tier evaluation protocol. During candidate search, each candidate pose is evaluated using `Vina.optimize()`, which performs local BFGS minimization from the pocket-aligned candidate pose and provides the online score used for ranking the candidate pool. Candidate poses are constructed by constrained RDKit embedding, where the retained molecular scaffold is aligned to the parent ligand pose. Only the selected challenger is subsequently evaluated using a fresh `Vina.dock()` global search. The challenger and current certified checkpoint are re-docked as a paired and uncached comparison before a commitment decision is made. This design prevents the optimizer from treating the inexpensive screening score as a final certificate of improvement.

In addition to docking affinity, we compute QED and normalized synthetic accessibility (SA) using RDKit-based molecular descriptors. These properties are used during multi-objective search but are not imposed as universal hard filters, allowing initially weak leads to be repaired during optimization.

A.3 Fragment Library and Molecular Edit Space

The fragment library contains 5,563 candidate fragments before filtering. We retain 1,164 fragments with at most six heavy atoms after removing reactive or undesirable substructures. Fragments are

categorized by heavy-atom count into SMALL, MEDIUM and LARGE groups, corresponding to 1–3, 4–6 and over 7 heavy atoms , respectively.

The optimizer operates through three local molecular transformations: ADD, which attaches a library fragment to an eligible heavy atom; REPLACE, which removes a peripheral substituent across an acyclic single bond and grafts a new fragment; and DELETE, which removes a peripheral substituent. Every generated molecule must pass RDKit sanitization and structural validity checks. Reactive substructures and heteroatom–halogen artifacts are removed, and each individual edit is restricted by the configured spatial edit radius and maximum growth constraint.

A.4 OPTIMIZATION STATE AND CANDIDATE SELECTION

At each optimization step, the agent maintains a planning state $z = (H_{\text{aff}}, H_{\text{local}}, H_{\text{broad}}, B_{\text{QED}}, B_{\text{SA}}, r)$, where H_{aff} denotes the remaining affinity deficit, H_{local} and H_{broad} summarize recently reachable improvement under local and broad search, B_{QED} and B_{SA} describe remaining molecular-property slack, and r denotes the current reliability state. The state determines whether the optimizer should prioritize affinity improvement, property repair, or balanced optimization.

We use the standard success thresholds $V_{\text{thr}} = -8.18$, $Q_{\text{thr}} = 0.25$, and $S_{\text{thr}} = 0.59$. The corresponding deficits are

$$d_V = \max(0, V - V_{\text{thr}}), \qquad d_Q = \max(0, Q_{\text{thr}} - \text{QED}), \qquad d_S = \max(0, S_{\text{thr}} - \text{SA}). \quad (6)$$

When affinity is the only unmet objective, candidate selection prioritizes docking improvement while maintaining QED and SA above their required thresholds. When molecular properties are deficient but affinity already satisfies the target, the optimizer prioritizes property repair while preserving the available affinity margin. When both affinity and molecular properties are deficient, candidates are ranked using a normalized multi-objective score with relative weights $W_V/W_Q/W_S = 2/1/0.5$.

Each optimization step initially evaluates $N = 40$ candidate edits. A candidate must improve the current fast-Vina score by at least 0.1 kcal/mol to be considered an affinity-improving challenger, except during explicit property-repair states where approximately neutral affinity changes are permitted. If no productive candidate is found, the proposal pool is enlarged by a factor of three before the trajectory is declared locally converged.

A.5 DECISION MODULE

The decision module does not directly generate a molecular structure. Instead, it defines a factorized stochastic distribution over high-level molecular-edit decisions. For each factor value k, corresponding to an edit operation, edit size, search scope, or editable region, we assign a proposal score

$$\begin{aligned} S_t(k) = O_t(k) &- \lambda_Q c_Q(k) - \lambda_S c_S(k) - \lambda_H c_H(k) \\ &+ \beta_U U_t(k) - \lambda_R(r_t)\rho_t(k) \\ &+ \log b_{\text{planner}}(k) + \lambda_{\text{LLM},t} \log b_{\text{LLM}}(k). \end{aligned} \quad (7)$$

where $O_t(k)$ is the target-local verified opportunity estimate, $c_Q(k)$, $c_S(k)$, and $c_H(k)$ denote molecular-property costs associated with QED, SA, and molecular growth, respectively, $U_t(k) = 1/\sqrt{n_k + 1}$ is an exploration bonus for under-sampled decisions, and $\rho_t(k)$ is the historical reliability-risk statistic. The terms $b_{\text{planner}}(k)$ and $b_{\text{LLM}}(k)$ provide deterministic state-dependent and bounded semantic preferences, respectively. In Eq. 1, the combined property cost is $C_t^{(d)}(k) = \lambda_Q c_Q(k) + \lambda_S c_S(k) + \lambda_H c_H(k)$ for the corresponding factor, and $b_{\text{struct},t}^{(d)}(k)$ denotes the deterministic prior written here as $b_{\text{planner}}(k)$.

The factor probabilities are obtained using an ϵ-mixed Boltzmann distribution,

$$\pi_t(k) = (1 - \epsilon)\frac{\exp(S_t(k)/T)}{\sum_{j \in \mathcal{K}} \exp(S_t(j)/T)} + \frac{\epsilon}{|\mathcal{K}|}, \quad (8)$$

followed by a maximum-probability cap and renormalization. The Boltzmann term favors high-scoring decisions while preserving stochastic exploration, whereas the uniform component prevents the search distribution from collapsing prematurely onto a single edit direction.

We use $\epsilon = 0.20$, $T = 0.70$, $\beta_U = 0.20$, and a maximum factor probability of 0.70. Each factor is sampled independently from its corresponding categorical distribution. Editable regions are sampled using the same memory-guided formulation over the valid sites of the current molecule. Once the high-level factors are sampled, the concrete fragment is drawn stochastically from the compatible fragment subset rather than being selected directly by the language model.

A.6 Certified Experience Memory

The target-local memory records statistics over edit operation, edit size, search scope, and molecular region. Positive opportunity statistics are updated only from outcomes that pass the independent certification procedure. A candidate that appears favorable according to the online screening score but fails certification therefore contributes a failure or reliability event rather than a positive experience. This separation prevents repeated proxy errors from becoming increasingly preferred search directions.

For each factor value k, the opportunity estimate combines the posterior probability of certified success with the expected positive certified gain. A Beta$(1, 1)$ prior is used for the success probability. Memory is reset at the beginning of every target and no optimization experience is transferred across test proteins.

A.7 Risk Control and Certification

The risk-control module separates candidate discovery from molecular commitment. For each selected challenger, we collect candidate-specific evidence including paired full-docking gain, pose consistency, screening–certification reversal, unusually large local gain, disagreement between affinity evaluators, auxiliary neural-model uncertainty, interaction changes, molecular-property changes, and recent certification history.

Risk evidence is divided into an evidence channel and an advisory chemistry channel. A RED state, which can block checkpoint replacement, requires a measured candidate-specific anomaly such as severe pose inconsistency, reversal between online screening and independent certification, repeated certification failure, or a high evidence concern corroborated by an observed anomaly. Disagreement from the auxiliary neural network affinity predictor or elevated MC-dropout uncertainty alone does not veto a Vina-certified improvement.

For affinity-optimization states, a challenger $x_t^\star$ replaces the certified checkpoint c_t only if

$$\mathrm{Commit}(x_t^\star) = \mathbb{I}\big[g_t^{\mathrm{cert}} > \delta_{\mathrm{cert}}\big]\, \mathbb{I}[\mathrm{Feasible}(x_t^\star)]\, \mathbb{I}[R_t \neq \mathrm{RED}]\,, \tag{9}$$

where $\delta_{\mathrm{cert}} = 0.05$ kcal/mol. Rejected challengers do not modify the checkpoint, and search resumes from the last certified molecular state. For property-repair states, a property-improving challenger may be committed when it preserves the required affinity margin.

A.8 Language-Model Components

All language-model components use a frozen claude-4.8-opus backend with temperature zero and structured JSON output. The LLM never emits a SMILES string, exact fragment identity, atom identifier, or final COMMIT/REJECT/STOP decision. Failure to produce a valid structured response after one repair attempt results in a neutral fallback.

The Pocket Planner receives deterministic descriptions of the pocket environment, molecular-property slack, affinity deficit, search headroom, risk state, and semantic memory. It returns only bounded categorical preferences over predefined edit factors. The values DISFAVOR, NEUTRAL, and PREFER are mapped to fixed multipliers 0.70, 1.00, and 1.35, respectively. The influence of the LLM decreases as certified target-specific evidence accumulates.

The Experience Summarizer periodically converts structured certified statistics into a compact semantic description of productive and unproductive search patterns. The Evidence Critic receives candidate-specific measurements and returns bounded evidence and chemistry concern levels. These outputs are advisory and cannot independently determine molecular commitment.

Table 3: **Default optimization hyperparameters.**

Hyperparameter	Value	Hyperparameter	Value
Candidates per step N	40	Exploration floor ϵ	0.20
Softmax temperature T	0.70	MC-dropout passes	5
Maximum factor probability	0.70	Resampling multiplier	$3\times$
LOCAL edit radius	0.52	BROAD edit radius	0.75
Maximum HA growth per edit	10	Minimum online Vina gain	0.10
Certification threshold	0.05	Maximum optimization steps	10
$W_V/W_Q/W_S$	$2/1/0.5$	Fragment-library size	1,164
V_{thr}	-8.18	$Q_{\mathrm{thr}}/S_{\mathrm{thr}}$	$0.25/0.59$
LLM $\lambda_{\max}$	1.0	LLM evidence scale τ	6

A.9 Evaluation Metrics

For every target, we report metrics for one delivered molecule. Vina denotes the full-docking score in kcal/mol and ΔVina measures the improvement relative to the initial reference ligand. We additionally report QED, normalized SA, heavy-atom change ΔHA, ligand efficiency $\mathrm{LE} = \mathrm{Vina}/N_{\mathrm{HA}}$, and the average number of optimization steps. Success Rate (SR) is the fraction of targets whose delivered molecules simultaneously satisfy

$$\mathrm{Vina} < -8.18, \qquad \mathrm{QED} > 0.25, \qquad \mathrm{SA} > 0.59. \tag{10}$$

Reporting ΔHA and ligand efficiency in addition to docking score is particularly important for molecular optimization because a method may obtain apparently favorable docking scores through unconstrained ligand growth without producing correspondingly efficient molecular improvements.

A.10 Ablations

We compare component variants that retain full-docking verification while removing selected control or proposal mechanisms. w/o Memory removes target-local experience terms from the proposal distribution. w/o Decision replaces adaptive factorized search with uniform sampling over edit operation, region, size, and scope and disables both the planner and memory-guided proposal bias. w/o Verification removes the evidence-based RED gate while retaining full-docking verification. This variant tests removal of the RED gate rather than predictive uncertainty in isolation.

A.11 Hyperparameters and Compute

All scientific models and tools remain frozen throughout test-time optimization; no parameter training or fine-tuning is performed on the CrossDocked2020 test targets. Targets are evaluated independently and can be parallelized across CPU workers. Runtime is dominated by Vina evaluations, with approximately 40 inexpensive online screens and a small number of full-docking certification calls per optimization step. Each trajectory is limited to at most 10 optimization steps, and Vina calls use per-target timeouts to prevent pathological docking failures from dominating the total runtime.

B LLM Prompts and Guardrails

Our framework uses claude-4.8-opus in three bounded and advisory roles: a Pocket Planner, an Experience Summarizer, and an Evidence Critic. All calls use temperature 0 and require structured JSON output. The language model never generates a molecule, SMILES string, fragment identifier, or atom identifier, and it cannot directly accept, reject, commit, or stop an optimization trajectory. These decisions remain under the deterministic proposal, certification, and risk-control mechanisms described in the main text. If an LLM response cannot be parsed or violates the required schema, the system uses a hard-coded neutral fallback, so an invalid model response reduces to "no preference" rather than changing the optimization state.

Across all three roles, the LLM is restricted to predefined categorical outputs and may use only measurements, region identifiers, residue identifiers, and statistics explicitly supplied in the input. It

is instructed not to infer unsupported interactions, geometry, protonation states, or molecular facts. We provide the system prompts below for reproducibility.

B.1 POCKET PLANNER

The Pocket Planner provides a bounded semantic prior over the predefined stochastic edit space. It receives the current optimization state, deterministic pocket and region descriptors, and accumulated semantic memory. Its output biases the probabilities of edit operation, fragment size, search scope, and editable region, but does not identify a concrete molecular edit. The categorical preferences PREFER, NEUTRAL, and DISFAVOR are mapped to fixed numerical multipliers 1.35, 1.00, and 0.70, respectively. The contribution of the LLM prior is further annealed as certified target-specific experience accumulates.

System prompt.

```
You are the Pocket Planner in a molecular optimization system.
Provide a BOUNDED SEMANTIC PRIOR over stochastic edit categories.

You do NOT generate molecules, SMILES, fragment/atom ids, or
exact edits, and you never accept/reject/commit/stop or override
constraints.

You may prefer/disfavor:
- operation class (ADD/REPLACE/DELETE),
- fragment size (SMALL/MEDIUM/LARGE),
- scope (LOCAL/BROAD),
- SUPPLIED region ids.

You may also set qualitative aggressiveness and property-spend.

Use ONLY supplied region ids and the allowed enums:
{DISFAVOR, NEUTRAL, PREFER},
aggressiveness {LOW, MEDIUM, HIGH},
property_spend {LOW, MEDIUM, HIGH},
confidence {LOW, MODERATE, HIGH}.

Do not invent residues/interactions/geometry.
If evidence is insufficient, return NEUTRAL biases and LOW
confidence.

Return STRICT JSON:
{
  "operation_bias": {
    "ADD": "...",
    "REPLACE": "...",
    "DELETE": "..."
  },
  "size_bias": {
    "SMALL": "...",
    "MEDIUM": "...",
    "LARGE": "..."
  },
  "scope_bias": {
    "LOCAL": "...",
    "BROAD": "..."
  },
  "region_bias": [
    {
      "region_id": "<supplied id>",
```

```
      "priority": "PREFER|NEUTRAL|DISFAVOR"
    }
  ],
  "aggressiveness": "LOW|MEDIUM|HIGH",
  "property_spend": {
    "QED": "...",
    "SA": "...",
    "SIZE": "..."
  },
  "confidence": "LOW|MODERATE|HIGH"
}
```

Input. The planner receives a structured state containing the current and target Vina values, affinity deficit, QED and SA values and remaining slack, heavy-atom growth relative to the starting ligand, recent local and broad search headroom, and the current risk state. For every editable region, the input additionally contains deterministic descriptors such as local environment class, free space, steric crowding, nearby residues, and the corresponding valid edit operation. Target-local semantic memory and recent certification statistics are included when available.

Use of the output. The planner output enters only as a soft bias in the factorized proposal distribution. In particular, the LLM cannot select a concrete fragment or molecular state. A disfavored option remains reachable because the proposal distribution retains a uniform exploration floor, and the influence of the LLM decreases as certified experience accumulates.

B.2 Experience Summarizer

The Experience Summarizer converts target-local certified statistics into a small set of structured search lessons. It does not evaluate molecules or choose subsequent edits. Importantly, only certified outcomes may support a positive lesson; favorable online screening scores that fail certification cannot be converted into successful experience.

System prompt.

```
You are the Experience Summarizer.

Convert CERTIFIED trajectory statistics into compact structured
lessons. You never choose the next edit or evaluate molecules.

Summarize ONLY patterns supported by the supplied certified stats.
Attach numeric support to every lesson.
Never treat uncertified online gains as success.
Do not invent chemistry.
No HIGH-confidence lesson from sparse data (<6 trials).
Use only supplied factor keys/region ids.

Return STRICT JSON:
{
  "productive_patterns": [
    {
      "scope": "OPERATION|SIZE|SCOPE|REGION",
      "target": "<key>",
      "statement": "<=20 words",
      "confidence": "LOW|MODERATE|HIGH",
      "support": {
        "n_trials": <int>,
        "n_cert_success": <int>,
        "mean_cert_gain": <float>
      }
    }
```

```
  ],
  "unproductive_patterns": [],
  "risk_patterns": []
}
```

Input. The summarizer receives the current optimization step together with target-local per-factor statistics over operation, fragment size, search scope, and molecular region. For each factor value, the structured input records the number of previous trials, number of certified successes, and mean certified improvement. These statistics originate from the deterministic experience memory rather than from free-form language-model observations.

Use of the output. The resulting lessons form the semantic-memory component supplied to subsequent Pocket Planner calls. Every lesson retains its numerical support, and high-confidence summaries are prohibited when fewer than six observations are available. The semantic summary therefore compresses existing certified evidence but cannot create new positive evidence.

B.3 Evidence Critic

The Evidence Critic evaluates whether the structured measurements associated with a proposed challenger contain evidence of unreliable evaluator behavior. It explicitly separates candidate-specific evidence of evaluator unreliability from generic chemical concern. The critic returns two bounded concern levels, but it cannot directly reject a candidate. A high evidence concern contributes to a RED reliability state only when corroborated by a measured anomaly.

System prompt.

```
You are an evidence-grounded risk critic for molecular
optimization.

Distinguish MEASURED evidence of evaluator unreliability from
generic chemistry concerns.

Do NOT label a candidate HIGH evidence risk merely because:
- the edit is large, polar, or unusual,
- it adds a heterocycle,
- it changes a pharmacophore,
- docking generally neglects desolvation or entropy.

Generic docking limitations are NOT candidate-specific evidence.

HIGH evidence_concern REQUIRES at least one candidate-specific
anomaly present in the supplied measurements, such as:
- evaluator direction conflict,
- pose inconsistency,
- certification reversal,
- interaction collapse,
- outlier gain corroborated by another signal,
- repeated certification failure,
- hard property violation.

A plausible but unverified chemical mechanism goes under
chemistry_concern, with evidence_concern LOW or MODERATE.

Large certified gain is NOT itself suspicious unless combined
with a measured inconsistency.

Use ONLY supplied measurements.
Do not invent interactions, protonation, or pose facts.
```

Table 4: **Robustness across LLM backends.** The optimization pipeline, scientific evaluators, and decision rules are held fixed while only the frozen LLM backend is changed. Results are reported as mean ± standard deviation over independent runs.

LLM Backend	Vina ↓	QED ↑	SA ↑	SR (%) ↑
Claude-4.8 Opus	-8.636 ± 0.066	0.526 ± 0.035	0.683 ± 0.023	60 ± 3
GPT-5.4	$\mathbf{-8.780 \pm 0.051}$	0.540 ± 0.035	$\mathbf{0.710 \pm 0.010}$	61 ± 1
GPT-OSS-120b	-8.680 ± 0.075	$\mathbf{0.560 \pm 0.037}$	$\mathbf{0.710 \pm 0.016}$	$\mathbf{62 \pm 4}$

```
Return JSON only:
{
  "evidence_concern": "LOW|MODERATE|HIGH",
  "chemistry_concern": "LOW|MODERATE|HIGH",
  "evidence_flags": [...],
  "chemistry_flags": [...],
  "note": "<=20 words"
}
```

Input. The critic receives only structured measurements of the current checkpoint and challenger, including their docking and molecular-property values, edit type and size, paired certification gain, pose consistency, evaluator-direction agreement, interaction retention, certification reversal, gain outlier statistics, recent certification failures, and the current reliability state.

Use of the output. The critic output is advisory and is consumed by a deterministic risk controller. A HIGH evidence concern may contribute to a RED state only when at least one candidate-specific measured anomaly is also present. A MODERATE evidence concern or a HIGH chemistry concern may raise the state to YELLOW, but chemistry concern alone can never produce RED. Thus, the language model is never the sole reason for rejecting a molecular candidate.

B.4 Failure Handling and Bounded Influence

All LLM calls use deterministic decoding with temperature 0 and role-specific token limits. Outputs are parsed against the expected structured schema, and malformed, incomplete, or invalid responses are replaced by role-specific neutral defaults. For the Pocket Planner this corresponds to neutral proposal biases, while for the Evidence Critic it corresponds to no additional evidence-based concern. Consequently, the LLM can alter soft search preferences and summarize certified experience, but it cannot bypass molecular validity constraints, independently certify an improvement, overwrite the current checkpoint, or terminate the optimization process.

B.5 Robustness to LLM Backends

To examine whether UMO-Agent depends strongly on a particular language model, we repeat the evaluation with different frozen LLM backends while keeping the optimization pipeline and scientific evaluators unchanged. As shown in Table 4, performance remains similar across Claude-4.8 Opus, GPT-5.4, and GPT-OSS. Joint success rates range only from 60% to 62%, while the corresponding Vina, QED, and SA values remain comparable. These results suggest that the performance of UMO-Agent is not highly sensitive to the specific LLM backend used in its bounded planning and critique modules.