Popper: Pareto-Guided Agentic Learning for Interpretable Peptide Lead Optimization

Ruochi Zhang[*1], Tian Wang[*1], Li Jiao[*1], Yusi Fan[2], Qiong Zhou[1], Mengyuan Ji[1], Jialei Liu[1], Lanxin Xing[1], Qian Yang[2], Fengfeng Zhou[2], Chang Liu[3], and Xin Gao[†4,5,6]

[1]Syneron Opal, 10281, Cayman Island
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun, Jilin, 130012, China
[3]Beijing Life Science Academy, Beijing, 102209, China
[4]Computer Science Program, Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia
[5]Center of Excellence for Smart Health (KCSH), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia
[6]Center of Excellence on Generative AI, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia

Abstract

Peptide lead optimization is a constrained, multi-objective, multi-round decision process conducted under weak, noisy, and partially missing feedback. The key challenge is to convert heterogeneous experimental and in silico signals into actionable local edits while respecting strict constraints and evaluation budgets. Existing generative approaches often underutilize project-specific structure–activity relationship (SAR) evidence, while heuristic optimizers lack auditable strategy adaptation. We present Popper, a Pareto-guided multi-agent agentic learning framework that treats a Large Language Model (LLM) as a reasoning orchestrator across specialized Insight, Design, and Evaluation agents. Popper enforces numeric governance by sourcing all quantitative signals from tools and oracles, while the LLM performs strategic hypothesis formation and reflection. The system couples: (i) interpretable, graph-based SAR rule mining with explicit evidence chains; (ii) a structured weak-feedback interface that aggregates multi-source utilities (potency, structural quality, energetics, developability) with explicit missingness handling; (iii) Pareto-based parent selection to preserve solution diversity; and (iv) structured reflection artifacts (JSON) that adaptively update optimization strategies across rounds. Experiments demonstrate that Popper improves sample efficiency, constraint satisfaction, and Pareto-front quality compared to state-of-the-art structure-based and sequence-based baselines, maintaining robust performance across diverse feedback regimes.

1 Introduction

Lead optimization fundamentally differs from de novo molecular generation [40, 9] in its iterative, hypothesis-driven nature. While generative models excel at sampling diverse molecular structures, lead optimization requires careful navigation of a constrained local search space, where each modification must be justified by evidence and evaluated against multiple competing objectives [23, 35, 4]. The iterative decision-making process mirrors the scientific method: hypotheses are formed based on structure–activity relationship (SAR) patterns, candidates are designed to test these hypotheses, and strategies are refined based on feedback. However, lead optimization operates under challenging conditions that distinguish it from well-controlled optimization problems. The operating environment is characterized

[*]These authors contributed equally to this work.
[†]Corresponding author: xin.gao@kaust.edu.sa

by small datasets with high noise, where limited experimental measurements and assay variability introduce significant uncertainty into feedback signals [16]. Multi-assay inconsistency further complicates the optimization landscape, as different assays may imply conflicting optimization directions—some objectives require minimization while others require maximization, creating complex trade-off surfaces that cannot be collapsed into a single scalar metric [10]. The expense of evaluation presents another critical constraint: structure prediction and energy scoring operations impose substantial computational and time budgets [21], making exhaustive search infeasible and necessitating budget-aware optimization strategies [36, 5]. Moreover, feedback signals are often incomplete or delayed, requiring robust handling of missing data and uncertainty propagation.

Traditional optimization approaches often treat lead optimization as a single-pass generation problem or employ black-box scoring functions that obscure the reasoning behind candidate selection. Multi-objective evolutionary algorithms [10] and Bayesian optimization methods [36] provide algorithmic foundations but lack domain-specific adaptations for lead optimization. They do not incorporate interpretable evidence substrates such as SAR rules that can guide candidate generation beyond pure numerical optimization, nor do they support agentic planning and explanation capabilities that enable human-in-the-loop optimization workflows [39]. Structure-based methods such as RFDiffusion combined with ProteinMPNN [40, 9] excel at generating structurally plausible candidates but typically underutilize project-specific SAR patterns and lack explicit mechanisms for constraint-governed multi-round adaptation where optimization strategies must evolve based on feedback from previous rounds.

We frame lead optimization as *agentic learning* under weak, structured feedback, treating the large language model (LLM) [6, 38, 11] as a reasoning orchestrator rather than a free-form generator. In this framework, the LLM emulates a medicinal chemist by forming hypotheses, planning constraint-aware local edits, and routing among specialized agents. We decompose the optimization task into three specialized roles—Insight, Design, and Evaluation—to prevent agent confusion and ensure that evidence generation, candidate proposal, and quantitative assessment remain semantically distinct. Critically, all quantitative signals originate from specialized tools and oracles to eliminate hallucinated numerical predictions, while strategy updates are driven by structured reflection artifacts that audit the reasoning process at each round. This bounded multi-agent architecture enables reproducible execution paths while leveraging the LLM's capacity for high-level strategic adaptation.

This work makes the following contributions:

1. **Agentic learning architecture:** We introduce a specialized Insight/Design/Evaluation framework that enforces semantic boundaries between evidence mining, strategic proposal, and rigorous assessment. This decomposition improves reproducibility and prevents the "agent confusion" often seen in monolithic systems, while ensuring all quantitative governance is tool-backed.

2. **Interpretable SAR rule-mining substrate:** We introduce a graph-based rule extraction system that builds composable mutation rules with quantified effects, constructs a rule-compatibility graph via additivity tests, and proposes candidates via clique search, transitive-clique expansion, and subtraction-based exploration, enabling the LLM to reason over SAR as structured external memory rather than unstructured correlations.

3. **Pareto-guided, multi-round optimization loop:** We present an optimization framework that selects parents from Pareto fronts using non-dominated sorting and crowding-distance diversity metrics, and adapts exploration versus exploitation through structured reflection with deterministic fallbacks and explicit convergence criteria, ensuring robust multi-objective optimization under uncertainty.

4. **Structured weak-feedback interface:** We design a hierarchical multi-objective scoring system that unifies multi-source signals (potency, structural quality, energetics, developability) into auditable outcomes with uncertainty and missingness handling, exposing per-candidate score breakdowns and evidence chains that enable transparent decision-making.

2 Related Work

2.1 Structure-/Sequence-Based Generation and Optimization

Structure-based binder design has advanced significantly through methods such as RFDiffusion [40] combined with ProteinMPNN [9], which leverage geometric and physical constraints to generate protein–peptide complexes. These approaches excel at producing structurally plausible candidates by incorporating folding constraints and interface geometry directly into the generation process [1, 20]. Sequence-based peptide language models and masked language

model (MLM) mutation sampling, such as PepMLM [2], provide alternative strategies that operate on sequence grammars and in silico evolution principles, enabling efficient exploration of sequence space while respecting biochemical constraints. Recent advances in protein language models [33, 28, 25] have demonstrated the power of large-scale sequence representations for structure and function prediction.

However, these methods miss critical aspects required for effective lead optimization in practice. They typically underutilize project-specific SAR patterns, treating each optimization task as an isolated generation problem rather than learning from accumulated experimental evidence. Moreover, they lack explicit mechanisms for constraint-governed multi-round adaptation, where optimization strategies must evolve based on feedback from previous rounds. The absence of auditable strategy updates makes it difficult to understand why certain candidates were selected and how the optimization process adapted to new information.

2.2 Multi-Objective Optimization under Expensive/Weak Feedback

Multi-objective evolutionary algorithms, particularly NSGA-II [10] and related Pareto-based optimizers [8, 19], provide well-established frameworks for handling conflicting objectives. These methods maintain diverse solution sets through non-dominated sorting and crowding distance metrics, enabling exploration of trade-off surfaces without premature convergence to suboptimal regions. Bayesian optimization [36, 5] and bandit-style methods offer complementary approaches for efficient exploration under limited feedback, using probabilistic models to guide sample selection and balance exploration with exploitation [32].

While these optimization frameworks provide solid algorithmic foundations, they lack domain-specific adaptations required for lead optimization. They do not incorporate interpretable evidence substrates such as SAR rules that can guide candidate generation beyond pure numerical optimization. The absence of structured feedback interfaces tied to real scientific signals means that these methods cannot leverage domain knowledge or provide explanations for their recommendations. Moreover, they do not support agentic planning and explanation capabilities that enable human-in-the-loop optimization workflows.

2.3 Bounded Agentic Loops

Recent advances in tool-augmented LLM agents [6, 38] have demonstrated the potential for combining language model reasoning with specialized computational tools. In scientific domains, these agents can orchestrate complex workflows involving multiple tools, dynamically routing tasks and integrating results [39]. However, a critical distinction exists between bounded loops with structured state and open-ended chat agents that lack explicit termination conditions and reproducible execution paths.

Our work contributes to this landscape by framing multi-round optimization as an explicit decision process with structured state, deterministic fallbacks, and auditable artifacts. Unlike open-ended conversational agents, our system operates within well-defined agent boundaries (Insight, Design, Evaluation) with clear semantic roles and explicit state transitions. This bounded architecture enables reproducibility and systematic evaluation while maintaining the flexibility of LLM-driven reasoning.

3 Problem Formulation

3.1 Optimization as Sequential Decision-Making

We model peptide lead optimization as a constrained, multi-objective, finite-horizon sequential decision process with partial and noisy feedback. Let Σ denote a finite residue alphabet and let Σ^L be the set of peptide sequences of fixed length L. The optimization proceeds in rounds $t \in \{0, 1, \ldots, T-1\}$.

State. The state s_t summarizes all information available at the beginning of round t, including a (multi-)parent set $\mathcal{P}_t \subseteq \Sigma^L$, accumulated evidence (e.g., mined SAR rules and trend summaries), and multi-round history. We treat s_t as a sufficient statistic for decision-making (implemented as an explicit, auditable state object).

Action (bounded local edits). An action a_t specifies a constrained local-edit policy that, when applied to the parent set, generates a candidate batch $\mathcal{C}_t \subseteq \Sigma^L$. Each candidate must satisfy hard feasibility constraints. Concretely, given protected positions $\mathcal{I}_{\text{prot}} \subseteq \{1, \ldots, L\}$ and a mutation budget $K \in \mathbb{N}$, the feasible set relative to $\mathcal{P}_t$ is

$$\mathcal{X}_{\text{feas}}(\mathcal{P}_t) \triangleq \left\{ x \in \Sigma^L \;:\; \exists p \in \mathcal{P}_t,\; \left|\{i : x_i \neq p_i\}\right| \leq K,\; x_i = p_i \;\forall i \in \mathcal{I}_{\text{prot}} \right\}, \tag{1}$$

optionally intersected with additional chemistry constraints (e.g., ring-closure requirements).

Multi-objective outcomes with missingness and noise. Evaluating a candidate x yields a vector outcome

$$o(x) \triangleq \left(o^{(1)}(x), \ldots, o^{(M)}(x)\right) \in \left(\mathbb{R} \cup \{\text{None}\}\right)^M, \tag{2}$$

where each component corresponds to an objective (e.g., potency, structural quality, developability) and may be missing due to expensive or unavailable oracles. Let $m^{(j)}(x) \in \{0, 1\}$ denote the missingness indicator for objective j (1 if missing). Observations are noisy:

$$o^{(j)}(x) = \tilde{o}^{(j)}(x) + \varepsilon^{(j)}(x), \qquad \mathbb{E}[\varepsilon^{(j)}(x)] = 0, \tag{3}$$

capturing assay variability and model uncertainty.

3.2 Objectives and Constraints

We seek to optimize M potentially conflicting objectives. For each objective $j \in \{1, \ldots, M\}$, we define a standardized utility function $u^{(j)} : \Sigma^L \to [0, 1]$ with "higher is better" semantics. Raw measurements that require minimization (e.g., Kd, IC50) are transformed via monotone mappings to this standardized scale.

In our implementation, we use $M = 3$ objectives, defining the objective vector as:

$$u(x) \triangleq \left(u^{(1)}(x), u^{(2)}(x), u^{(3)}(x)\right) = \left(u_{\text{pot}}(x), u_{\text{struct}}(x), u_{\text{dev}}(x)\right), \tag{4}$$

where $u_{\text{pot}}(x)$, $u_{\text{struct}}(x)$, and $u_{\text{dev}}(x)$ denote potency, structural quality, and developability scores, respectively. Each component is computed by a structured, missingness-aware scoring interface (detailed in the Method section), with deterministic defaults applied when an oracle is unavailable.

Primary scalar target used for progress and stopping. In addition to the vector objectives, the workflow defines a scalar composite score $s(x) \in [0, 1]$ (reported as `final_score`) by hierarchically aggregating objective components and applying soft-constraint penalties (Method section). Multi-round convergence and exploration control use improvements in the best observed scalar score, $\max_{x \in \mathcal{C}_t} s(x)$, as the primary progress signal.

Pareto dominance. For two candidates $x, y \in \Sigma^L$ with (imputed) utility vectors $u(x), u(y) \in [0, 1]^M$, we say x *dominates* y (written $x \prec y$) if $u^{(j)}(x) \geq u^{(j)}(y)$ for all j and strict inequality holds for at least one objective. The (round-wise) Pareto set is the set of non-dominated candidates within the evaluated batch.

Hard vs. soft constraints. Hard constraints define feasibility, e.g., protected positions and mutation budgets as in $\mathcal{X}_{\text{feas}}(\mathcal{P}_t)$, and optional chemistry constraints for cyclic peptides. Soft constraints are represented as penalties that down-weight otherwise-feasible candidates (e.g., excessive net charge, high aggregation risk, low backbone confidence). This separation keeps constraint governance explicit and auditable.

3.3 Feedback as a Learning Signal

Feedback is weak in the sense that it is local (available only for proposed candidates), partially observed, and noisy. The learning signal at round t is the structured record $\{(x, o(x), m(x)) : x \in \mathcal{C}_t\}$ together with provenance metadata (tool versions, evidence chains, and uncertainty estimates). The agent updates its internal strategy (e.g., exploration/exploitation balance and rule preferences) from this record while maintaining interpretability via explicit, machine-readable artifacts.

3.4 Formal Objects

For completeness, the round-t interaction tuple is

$$\big(s_t,\ a_t,\ \mathcal{C}_t,\ \{o(x), m(x)\}_{x\in\mathcal{C}_t},\ c_t\big), \tag{5}$$

where $c_t \geq 0$ denotes the incurred evaluation cost (e.g., number of expensive oracle calls such as structure prediction and energy scoring). A global budget constraint can be written as $\sum_{t=0}^{T-1} c_t \leq B$. Consistent with our implementation, we treat optimization as a cost-constrained, batch, black-box search problem under partial and noisy evaluations: we aim to discover high-scoring candidates in terms of the scalar score $s(x)$ while maintaining a diverse set of non-dominated solutions under the vector objectives $u(x)$ (used for Pareto-based parent selection).

4 Method

4.1 Algorithmic Overview

We frame lead optimization as a bounded, multi-agent collaborative decision process operating under weak feedback. The system implements a LangGraph [22] state machine with three specialized agents $\mathcal{A} = \{\text{Insight}, \text{Design}, \text{Evaluation}\}$ that collaborate through an orchestrator to solve different aspects of lead optimization. The core algorithmic components are: (1) an interpretable SAR rule-mining substrate that extracts composable mutation rules with quantified effects from experimental data; (2) a hierarchical multi-objective scoring interface that aggregates heterogeneous signals (potency, structure, energetics, developability) into auditable outcomes with explicit missingness and uncertainty handling; (3) a Pareto-guided multi-round optimization loop that selects parents via non-dominated sorting, adapts exploration/exploitation balance through structured reflection, and detects convergence via plateau-based criteria; and (4) a constraint-aware candidate generation mechanism that combines rule-based proposals, SAR-guided LLM edits, and novel medicinal chemistry designs.

The workflow begins with intent classification $\pi_{\text{intent}} : \mathcal{U} \to \mathcal{A}$ that maps user queries $\mathcal{U}$ to appropriate agents. Each agent operates as a bounded subgraph with deterministic state transitions, explicit termination conditions, and auditable artifacts. Agent coordination is managed by an orchestrator that routes tasks and maintains shared state through explicit interfaces [22]. All quantitative outputs originate from specialized tools (ML models [24, 13], structure predictors [17, 3], energy scorers) rather than LLM hallucinations, ensuring reproducibility and numerical correctness. Figure 1 illustrates the overall system architecture and workflow.

4.2 Multi-Agent Collaborative Architecture

Agent Composition and Orchestration. The system implements a multi-agent collaborative architecture where specialized agents coordinate to solve different aspects of lead optimization. The architecture consists of three primary agents: an *Insight Agent* for SAR analysis and evidence extraction, a *Design Agent* for candidate generation and multi-round optimization, and an *Evaluation Agent* for external candidate assessment. This decomposition isolates specialized reasoning tasks, preventing the performance degradation often observed when a single LLM context is tasked with simultaneous evidence analysis and candidate design. Each agent $a \in \mathcal{A}$ operates with a dedicated state space S_a, action set A_a, deterministic transition functions $T_a : S_a \times A_a \to S_a$, and output generation functions $R_a : S_a \to \mathcal{O}_a$. Agent coordination is managed by an orchestrator that routes tasks based on user intent and maintains shared state through explicit interfaces, preventing state leakage while enabling information flow between agents.

Insight Agent. The Insight Agent operates on dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where x_i are peptide sequences and y_i are activity measurements. The agent provides two execution paths: a *light path* that performs fast SAR trend analysis via position-wise statistics, and a *full path* that trains a tabular ML model $f_\theta : \mathcal{X} \to \mathcal{Y}$ using deep learning techniques [24, 13, 18], computes SHAP [27, 31] feature importances, and performs comprehensive rule mining. The light path executes in $O(N \cdot L)$ time where L is sequence length, while the full path requires $O(N^2 \cdot L + T_{\text{train}})$ where T_{train} is model training time. The agent's outputs (SAR findings, mined rules, SHAP explanations) are shared with the Design Agent to guide candidate generation.

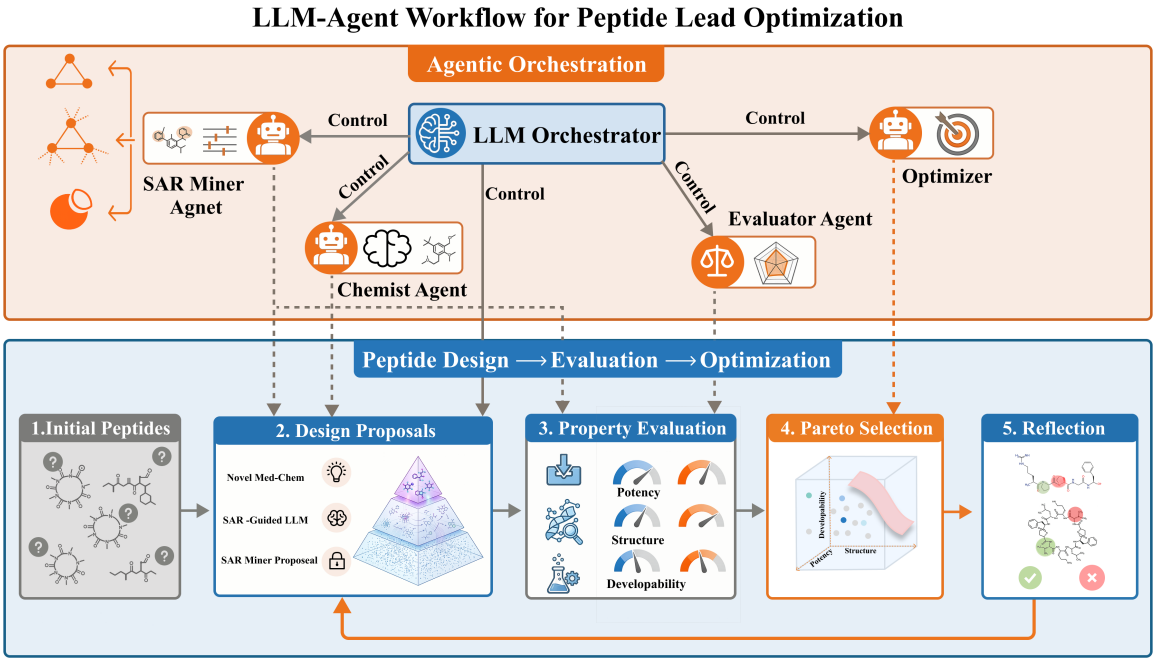


Figure 1: System overview. The workflow is structured into explicit agents (Insight, Design, Evaluation) with clear semantic boundaries. The LLM acts as a chemist-like reasoning orchestrator that routes among specialized components and proposes constraint-aware edits, while all quantitative signals are produced by tools (potency model, Boltz-2 structure, and energy scoring), yielding auditable per-round artifacts.

Design Agent. The Design Agent implements a bounded multi-round optimization loop $\mathcal{L}_{\text{design}}$ that iteratively generates, scores, and refines candidates. The loop state s_t at round t contains: parent sequences $\mathcal{P}_t \subseteq \Sigma^L$, accumulated SAR evidence $\mathcal{E}_t$ (mined rules, trend summaries) received from the Insight Agent, and optimization history $\mathcal{H}_t$. Each round executes: candidate generation $\mathcal{C}_t = \text{Generate}(s_t, \mathcal{E}_t, \text{constraints})$, multi-objective scoring $u(\mathcal{C}_t)$ via specialized tools (ML models, structure predictors, energy scorers), Pareto-based parent selection $\mathcal{P}_{t+1} = \text{SelectPareto}(\mathcal{C}_t, k)$, and strategy update via structured reflection $\mathcal{E}_{t+1} = \text{Reflect}(\mathcal{C}_t, s_t)$. The loop terminates when convergence criteria are met or maximum rounds T are reached.

Evaluation Agent. The Evaluation Agent takes a set of external candidates $\mathcal{C}_{\text{ext}}$ and produces structured assessments with risk flags, score breakdowns, and go/no-go recommendations. The agent collaborates with the same specialized tools used by the Design Agent (ML models, structure predictors, energy scorers), ensuring scoring consistency across the system. The agent's assessments can inform Design Agent decisions or support external validation processes.

Bounded Execution Guarantees. The architecture enforces boundedness through explicit termination conditions (max rounds, convergence thresholds, time limits), deterministic fallbacks for LLM parsing failures, tool call timeouts and retry policies, and state validation at each transition. All state updates are logged with provenance metadata, enabling complete audit trails. The orchestrator ensures that agent interactions remain within well-defined boundaries, preventing infinite loops and ensuring reproducible execution paths.

4.3 Interpretable SAR Rule Mining

Our SAR rule mining algorithm extracts composable mutation rules with quantified effects from experimental data, providing an interpretable evidence substrate for candidate generation. Unlike black-box QSAR models [7, 34, 29], our approach produces explicit rules with effect magnitudes and compatibility relationships, enabling systematic exploration of structure–activity space.

Rule Extraction. The algorithm operates on dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where $x_i \in \Sigma^L$ are peptide sequences of fixed length L over residue alphabet Σ, and $y_i \in \mathbb{R}$ are observed activity values. Each sequence is represented as a position-wise mapping $x : \mathcal{P} \to \Sigma$ where $\mathcal{P} = \{1, 2, \ldots, L\}$ is the set of positions.

For a wild-type sequence w and mutant m, we define the mutation set as $\Delta(w, m) = \{p \in \mathcal{P} : w(p) \neq m(p)\}$. A k-point mutation rule r is encoded as an ordered set of position-level mutations: $r = \{(p_i, a_i, b_i)\}_{i=1}^k$ where $a_i = w(p_i)$ and $b_i = m(p_i)$ are the wild-type and mutant residues at position p_i, with positions sorted: $p_1 < p_2 < \cdots < p_k$.

For rule r observed between wild-type w and mutant m, we define the *amplification factor* as:

$$a(r; w \to m) = \frac{f(m)}{f(w)} \tag{6}$$

where $f : \Sigma^L \to \mathbb{R}_+$ is a fitness function with "higher is better" semantics (e.g., $f(x) = 1/(\text{IC50}(x) \times 10^{-9})$ for IC50 in nM). The algorithm retains only rules satisfying $a(r; w \to m) \geq \alpha_{\text{th}}$ where α_{th} is a user-defined threshold (typically $\alpha_{\text{th}} \geq 1$ to focus on beneficial mutations).

Multiplicative Compositionality Assumption. The central modeling assumption is that rule effects compose multiplicatively when mutations are non-interacting:

$$a(r_1 \cup r_2) \approx a(r_1) \cdot a(r_2) \quad \text{if } r_1 \cap r_2 = \varnothing \tag{7}$$

This assumption enables prediction of combined rule effects through multiplication, provided the rules are validated as additive-compatible.

Additivity Testing. For a multi-point rule R with $|R| = k \geq 2$, we enumerate all non-trivial partitions (R_1, R_2) where $R = R_1 \cup R_2$, $R_1 \cap R_2 = \varnothing$, and both R_1 and R_2 are non-empty. For each partition, we compute the *relative factorization error*:

$$\epsilon(R; R_1, R_2) = \left| \frac{a(R)}{a(R_1) \cdot a(R_2)} - 1 \right| \tag{8}$$

If both R_1 and R_2 are observed rules (exist in the rule set $\mathcal{R}$) and $\epsilon(R; R_1, R_2) \leq \tau$ where τ is a tolerance parameter (default: 0.2), then (R_1, R_2) is marked as an *additive relation*. The algorithm constructs a *rule-compatibility graph* $G = (V, E)$ where vertices $V = \mathcal{R}$ represent all observed rules, and edges $(r_i, r_j) \in E$ exist if the pair participates in at least one validated additive relation. This graph structure enables efficient identification of compatible rule combinations through clique enumeration.

Candidate Generation Strategies. The algorithm employs three strategies of increasing exploration aggressiveness, each producing candidates with associated fitness predictions and evidence chains.

Strategy 1: Clique (Conservative). For each clique $C = \{r_1, r_2, \ldots, r_k\}$ in graph G where $k \geq 3$, we generate candidate $\hat{x} = \text{Apply}(w, C)$ by applying all rules in C to wild-type w. The predicted fitness uses a conservative lower bound:

$$\hat{f}_{\text{lb}}(\hat{x}) = f(w) \cdot a(r_1) \cdot \prod_{i=2}^{k} ((1 - \tau) \cdot a(r_i)) \tag{9}$$

where the $(1 - \tau)$ factor accounts for potential sub-additivity. This strategy only explores rule combinations that are directly validated by experimental data.

Strategy 2: TransitiveClique (Moderate Exploration). We expand graph G through limited transitive closure [8]: for each path $r_1 \to r_2 \to r_3$ where $(r_1, r_2) \in E$ and $(r_2, r_3) \in E$, we add edge (r_1, r_3) if there are no position conflicts (i.e., r_1 and r_3 do not mutate the same position to different residues). This process is repeated for max_hop iterations (default: 1). Cliques in the expanded graph are then used to generate candidates with predictions scaled by a decay factor:

$$\hat{f}(\hat{x}) = f(w) \cdot \beta \cdot \prod_{i=1}^{k} a(r_i) \tag{10}$$

where $\beta \in (0, 1]$ is a decay factor (default: 0.7) that accounts for additional uncertainty from transitive inference.

Strategy 3: Subtraction (Aggressive Exploration). This strategy uses *deduced rules* obtained through multiplicative factorization. For an observed multi-point rule $R = R_1 \cup R_2$ with amplification factor $a(R)$, if sub-rule R_1 is also observed with $a(R_1)$, we deduce the latent effect of the unobserved component R_2 as:

$$\hat{a}(R_2) = \frac{a(R)}{a(R_1)} \cdot \kappa \tag{11}$$

where κ is a correction factor (default: 1.0) accounting for potential interactions. Deduced rules $\hat{r}$ are retained if $|\hat{r}| \geq$ num_mut_min (default: 3) and $\hat{a}(\hat{r}) \geq$ amp_min (default: 10). Candidates are generated by applying deduced rules: $\hat{x} = \text{Apply}(w, \{\hat{r}\})$ with predicted fitness $\hat{f}(\hat{x}) = f(w) \cdot \hat{a}(\hat{r})$. This strategy enables the system to "subtract" known beneficial components from complex mutations, isolating and testing novel motifs that were not directly observed.

Evidence Chains and Provenance. Each generated candidate $\hat{x}$ is associated with an *evidence chain* that traces its provenance: the wild-type w, the set of rules R used to generate it, the strategy employed, and the nearest neighbors in dataset $\mathcal{D}$ (measured by sequence distance). This provenance information enables audit trails that link candidate proposals back to supporting experimental data, supporting interpretability and trust in the optimization process.

4.4 Hierarchical Multi-Objective Scoring

Our structured feedback interface implements a three-layer hierarchical scoring system that aggregates heterogeneous signals from multiple sources (ML predictions, structure prediction, energy scoring, developability assessment) into auditable, interpretable outcomes. The hierarchical design enables fine-grained control over component contributions while maintaining interpretability through explicit score breakdowns.

Layer 3: Interface Quality Aggregation. The interface quality score $u_{\text{interface}}(x) \in [0, 1]$ combines three subcomponents:

$$u_{\text{interface}}(x) = u_{\text{conf}}(x) \cdot w_{\text{conf}} + u_{\text{geom}}(x) \cdot w_{\text{geom}} + u_{\text{energ}}(x) \cdot w_{\text{energ}} \tag{12}$$

where $w_{\text{conf}} + w_{\text{geom}} + w_{\text{energ}} = 1$ (default: 0.30, 0.40, 0.30). The **interface confidence** component $u_{\text{conf}}(x) = \text{iPTM}(x)$ is the interface predicted TM-score from structure prediction [17, 3, 42], directly measuring interface region reliability. The **interface geometry** component $u_{\text{geom}}(x) = \frac{1}{2}(\text{interface\_sc}(x) + \text{interface\_packstat}(x))$ averages interface score and packing statistics, both normalized to $[0, 1]$. The **interface energetics** component $u_{\text{energ}}(x) = \frac{1}{3}\left(\tilde{u}_{\text{binder}}(x) + \tilde{u}_{\Delta G}(x) + \tilde{u}_{\text{hbonds}}(x)\right)$ averages three normalized energy metrics:

$$\tilde{u}_{\text{binder}}(x) = \frac{1}{1 + \text{binder\_score}(x)/s_{\text{REU}}} \tag{13}$$

$$\tilde{u}_{\Delta G}(x) = \frac{1}{1 + \text{interface\_dG}(x)/s_{\text{kcal}}} \tag{14}$$

$$\tilde{u}_{\text{hbonds}}(x) = \min\left(1.0, \frac{\text{interface\_hbonds}(x)}{h_{\max}}\right) \tag{15}$$

where $s_{\text{REU}} = 150$ REU, $s_{\text{kcal}} = 100$ kcal/mol, and $h_{\max} = 10$ are normalization scales. These transformations convert "lower is better" metrics (binder score, ΔG) and "higher is better" metrics (hydrogen bonds) into unified $[0, 1]$ scores.

Layer 2: Structural Quality Aggregation. The structural quality score combines interface and backbone metrics:

$$u_{\text{struct}}(x) = u_{\text{interface}}(x) \cdot w_{\text{interface}} + u_{\text{backbone}}(x) \cdot (1 - w_{\text{interface}}) \tag{16}$$

where $w_{\text{interface}} = 0.60$ (default) and $u_{\text{backbone}}(x) = \text{normalize}(\text{complex\_pLDDT}(x))$ is the normalized backbone confidence score from structure prediction [17, 21].

Layer 1: Base Score Computation. The base score aggregates all major objectives:

$$u_{\text{base}}(x) = u_{\text{pot}}(x) \cdot w_{\text{pot}} + u_{\text{struct}}(x) \cdot w_{\text{struct}} + u_{\text{dev}}(x) \cdot w_{\text{dev}} \tag{17}$$

where $w_{\text{pot}} + w_{\text{struct}} + w_{\text{dev}} = 1$ (default: 0.20, 0.60, 0.20). The **potency score** $u_{\text{pot}}(x) = \frac{1}{1+\text{KD}(x)/s_{\text{nM}}}$ uses the predicted binding affinity $\text{KD}(x)$ (nM) from the ML model, normalized by scale $s_{\text{nM}} = 1000$ nM. The **structural quality** component $u_{\text{struct}}(x)$ is computed from Layer 2. The **developability score** $u_{\text{dev}}(x) = \max(0, 1 - |\text{net\_charge}(x)|/10)$ penalizes high net charge, where net charge is computed as $\text{net\_charge}(x) = \sum_i \text{charge}(x_i)$ over all residues, following drug-likeness principles [26, 14].

Penalty Mechanism. Soft constraints are enforced through multiplicative penalties applied to the base score. The penalty factor $\phi(x) \in [\phi_{\min}, 1.0]$ aggregates four components:

$$\phi(x) = \max\left(\phi_{\min}, 1.0 - p_{\text{backbone}}(x) - p_{\text{charge}}(x) - p_{\text{aggregation}}(x) - p_{\text{seq}}(x)\right) \tag{18}$$

where $\phi_{\min} = 0.5$ is the minimum penalty factor. Each component uses indicator functions $\mathbb{1}[\cdot]$:

$$p_{\text{backbone}}(x) = \alpha_{\text{bb}} \cdot \mathbb{1}[u_{\text{backbone}}(x) < \theta_{\text{bb}}] \cdot (\theta_{\text{bb}} - u_{\text{backbone}}(x)) \tag{19}$$

$$p_{\text{charge}}(x) = \alpha_{\text{ch}} \cdot \mathbb{1}[\text{total\_charge}(x) > \theta_{\text{ch}}] \cdot (\text{total\_charge}(x) - \theta_{\text{ch}}) \tag{20}$$

$$p_{\text{aggregation}}(x) = \alpha_{\text{agg}}^{\text{high}} \cdot \mathbb{1}[\text{risk}(x) = \text{"high"}] + \alpha_{\text{agg}}^{\text{med}} \cdot \mathbb{1}[\text{risk}(x) = \text{"medium"}] \tag{21}$$

$$p_{\text{seq}}(x) = \alpha_{\text{run}} \cdot \text{max\_run}(x) + \alpha_{\text{ala}} \cdot \text{ala\_excess}(x) \tag{22}$$

with constants: $\alpha_{\text{bb}} = 0.2$, $\alpha_{\text{ch}} = 0.05$, $\alpha_{\text{agg}}^{\text{high}} = 0.15$, $\alpha_{\text{agg}}^{\text{med}} = 0.05$, $\theta_{\text{bb}} = 0.5$, $\theta_{\text{ch}} = 4$. Sequence penalties $p_{\text{seq}}(x)$ account for medicinal chemistry heuristics such as avoiding long hydrophobic runs or excessive alanine substitution.

Consistency Penalty (Multi-sample Structure Prediction). When multiple structure predictions are performed for a candidate x (multi-sample prediction), the system computes the standard deviation of pairwise RMSDs, denoted $\sigma_{\text{RMSD}}(x)$, to quantify prediction uncertainty. The consistency factor $\psi(x) \in [\psi_{\min}, 1.0]$ is computed via linear interpolation:

$$\psi(x) = \begin{cases} 1.0 - \frac{\sigma_{\text{RMSD}}(x)}{s_{\text{scale}}} \cdot (1.0 - \psi_{\min}) & \text{if } \sigma_{\text{RMSD}}(x) < s_{\text{scale}} \\ \psi_{\min} & \text{otherwise} \end{cases} \tag{23}$$

where $s_{\text{scale}} = 3.0$ Å (suitable for 12-mer peptides) and $\psi_{\min} = 0.1$. This ensures that highly inconsistent (and thus unreliable) structure predictions are down-weighted.

Final Composite Score. The final score $s(x)$ is the product of the base score and all penalty factors:

$$s(x) = u_{\text{base}}(x) \cdot \phi(x) \cdot \psi(x) \tag{24}$$

The product form ensures that problematic candidates in terms of either developability or structural reliability are prioritized lower, regardless of their predicted potency. The lower bound $\phi(x) \geq \phi_{\min}$ ensures that even heavily penalized candidates retain ranking signal, preventing complete exclusion while still down-weighting problematic designs.

Missingness and Uncertainty Handling. The interface explicitly represents missing values and uncertainty. For candidate x, each metric $m \in \mathcal{M}$ may be missing (denoted $m(x) = \text{None}$) due to tool unavailability or evaluation budget constraints. Missing metrics are handled through deterministic fallbacks:

$$u_m(x) = \begin{cases} u_m^{\text{observed}}(x) & \text{if } m(x) \neq \text{None} \\ u_m^{\text{default}} & \text{otherwise} \end{cases} \tag{25}$$

where $u_m^{\text{default}} = 0.5$ (neutral score) for most metrics, except $u_{\text{backbone}}^{\text{default}} = 0.7$ (optimistic default for structure). This explicit missingness representation enables the system to operate gracefully under partial feedback, a critical requirement for early-stage optimization where structure prediction may be too expensive for every candidate.

Pareto Objectives. For multi-objective optimization, the system defines a vector objective function $\mathbf{u}(x) = (u_{\text{pot}}(x), u_{\text{struct}}(x), u_{\text{dev}}(x)) \in [0,1]^3$ used for Pareto dominance calculations. This vector is computed independently of the scalar composite score $s(x)$, enabling Pareto-based parent selection that preserves diversity across objectives while the composite score guides convergence detection and exploration/exploitation balance.

4.5 Pareto-Guided Multi-Round Optimization

The multi-round optimization algorithm implements Pareto-guided parent selection, convergence detection, and dynamic exploration/exploitation balancing through structured reflection. Algorithm 1 presents the complete optimization loop.

Pareto-Based Parent Selection. At each round t, after scoring candidates $\mathcal{C}_t$, we select parents $\mathcal{P}_{t+1}$ for the next round using Pareto front optimization to prevent objective deterioration. The algorithm computes non-dominated fronts using fast non-dominated sorting (NSGA-II [10]).

For candidate x with objective vector $\mathbf{u}(x) = (u^{(1)}(x), \ldots, u^{(M)}(x))$, we say x *dominates* y (written $x \prec y$) if $u^{(j)}(x) \geq u^{(j)}(y)$ for all $j \in \{1, \ldots, M\}$ and strict inequality holds for at least one objective. Fast non-dominated sorting partitions candidates into fronts $\mathcal{F}_0, \mathcal{F}_1, \ldots$ where front $\mathcal{F}_0$ contains all non-dominated candidates (Pareto front), and front $\mathcal{F}_i$ contains candidates dominated only by fronts $\mathcal{F}_0, \ldots, \mathcal{F}_{i-1}$. The algorithm operates in $O(M \cdot N^2)$ time where M is the number of objectives and $N = |\mathcal{C}_t|$ is the number of candidates. For each candidate p, we maintain a domination count $d_p = |\{q \in \mathcal{C}_t : q \prec p\}|$ and a set $S_p = \{q \in \mathcal{C}_t : p \prec q\}$. Front $\mathcal{F}_0$ contains all candidates with $d_p = 0$. Subsequent fronts are built iteratively: for each candidate $p \in \mathcal{F}_i$, we decrement d_q for all $q \in S_p$, and candidates with $d_q = 0$ after this process form $\mathcal{F}_{i+1}$.

Crowding Distance for Diversity. When the first Pareto front $|\mathcal{F}_0| > k$ where k is the desired number of parents, we select diverse representatives using crowding distance. For each objective m, candidates are sorted by $u^{(m)}(\cdot)$, and boundary solutions (minimum and maximum) receive infinite crowding distance. For intermediate candidate i, the crowding distance in objective m is:

$$\text{cd}_i^{(m)} = \frac{u^{(m)}(x_{i+1}) - u^{(m)}(x_{i-1})}{u_{\max}^{(m)} - u_{\min}^{(m)}} \tag{26}$$

where $u_{\max}^{(m)}$ and $u_{\min}^{(m)}$ are the maximum and minimum values in the front. Total crowding distance is $\text{cd}_i = \sum_{m=1}^{M} \text{cd}_i^{(m)}$. We select the k candidates with highest crowding distance, ensuring diversity across the objective space.

If Pareto sorting fails (e.g., missing objective values), the system falls back to composite score ranking: $\mathcal{P}_{t+1} = \text{TopK}(\mathcal{C}_t, k, s(\cdot))$ where $s(x)$ is the scalar composite score.

Convergence Detection. The algorithm employs two termination mechanisms. First, the algorithm stops if $t \geq T_{\max}$ where $T_{\max}$ is the maximum number of rounds (default: 5). Second, plateau detection monitors the score improvement fraction $\delta_t = \frac{s_t - s_{t-1}}{|s_{t-1}|}$ where $s_t = \max_{x \in \mathcal{C}_t} s(x)$ is the top composite score at round t. The algorithm stops if $\delta_t < \theta$ for τ consecutive rounds, where $\theta = 0.01$ is the convergence threshold and $\tau = 3$ is the plateau patience. This mechanism prevents premature stopping due to temporary fluctuations while detecting true convergence.

Structured Reflection and Strategy Adaptation. After each round, the system performs structured reflection to update the optimization strategy. The reflection process analyzes top candidates $\mathcal{C}_t^{\text{top}} = \text{TopK}(\mathcal{C}_t, k_{\text{top}}, s(\cdot))$ and bottom candidates $\mathcal{C}_t^{\text{bottom}} = \text{BottomK}(\mathcal{C}_t, k_{\text{bottom}}, s(\cdot))$ to identify validated hypotheses (mutation patterns that appear frequently in top candidates), failed hypotheses (mutation patterns that appear frequently in bottom candidates), next-round strategy (actionable guidance for mutation planning), and the exploration ratio $\rho_{t+1} \in [0,1]$ that balances exploration (novel designs) versus exploitation (refining successful patterns). The reflection output is a structured JSON artifact (Listing 2) that is fed into the next round's mutation planning.

Dynamic Exploration/Exploitation Balancing. The exploration ratio ρ_t controls the fraction of candidates generated through novel medicinal chemistry proposals versus evidence-guided proposals. When LLM-based reflection succeeds,

Listing 1: Representative Reflection JSON Artifact

```
{
  "validated_hypotheses": [
    "P1 aromatic mutations (Phe, Trp) enhance interface packing and potency (+18%)",
    "C-terminal Gly insertion maintains backbone flexibility (iPTM > 0.82)"
  ],
  "failed_hypotheses": [
    "P4 charged mutations (Asp) increased solvent exposure but weakened binder_score",
    "P10-12 hydrophobic cluster destabilized the complex (pLDDT < 0.6)"
  ],
  "next_round_strategy": "Combine P1W with optimized C-terminal linker; prioritize
      hydrophobic substitutions at position 7 while keeping P4 neutral.",
  "exploration_ratio": 0.35,
  "reasoning": "Strong improvement (+12%) in round 2 suggests partial strategy validation;
      reducing exploration to refine promising aromatic clusters at the interface."
}
```

Figure 2: Representative reflection JSON artifact used to update next-round strategy.

ρ_{t+1} is determined by the LLM's analysis. When reflection fails (parsing errors, malformed JSON), a deterministic fallback computes ρ_{t+1} based on score improvement trends:

$$\rho_{t+1} = \text{clamp}\left(\rho_{\text{base}} + \Delta(\delta_t), 0, 1\right) \tag{27}$$

where $\rho_{\text{base}} = 0.4$ is the default exploration ratio and $\Delta(\delta_t)$ is the adjustment:

$$\Delta(\delta_t) = \begin{cases} -0.2 & \text{if } \delta_t > 0.05 \quad \text{(strong improvement} \rightarrow \text{exploit)} \\ -0.1 & \text{if } 0.01 < \delta_t \leq 0.05 \quad \text{(good improvement} \rightarrow \text{slight exploit)} \\ +0.1 & \text{if } 0 \leq \delta_t \leq 0.01 \quad \text{(weak improvement} \rightarrow \text{explore)} \\ +0.3 & \text{if } \delta_t < 0 \quad \text{(negative improvement} \rightarrow \text{strong explore)} \end{cases} \tag{28}$$

This adaptive mechanism automatically increases exploration when optimization stalls and decreases it when progress is strong, enabling robust optimization across diverse feedback regimes.

4.6 Constraint-Aware Candidate Generation

Candidate generation combines multiple proposal sources under hard and soft constraint satisfaction. The generation process takes as input: parent sequences $\mathcal{P}_t \subseteq \Sigma^L$, SAR evidence $\mathcal{E}_t$ (mined rules, trend analyses, SHAP [27] feature importances), structure and energy guidance from previous rounds, and constraints $\mathcal{C} = (\mathcal{I}_{\text{prot}}, K, \mathcal{V})$ where $\mathcal{I}_{\text{prot}} \subseteq \{1, \ldots, L\}$ are protected positions, $K \in \mathbb{N}$ is the maximum mutation budget, and $\mathcal{V} \subseteq \Sigma$ is the allowed amino acid vocabulary.

Three-Layer Proposal Architecture. The system employs a three-layer architecture that balances evidence-grounded proposals with exploratory designs:

Layer 1: SAR-Top Proposals (Exploitation). Directly applies top-ranked rules from the SAR rule-mining substrate. For each parent $p \in \mathcal{P}_t$, we select high-confidence rules $R_{\text{top}} \subseteq \mathcal{R}$ (e.g., rules with highest amplification factors or validated in cliques) and generate candidates $\mathcal{C}_1 = \{\text{Apply}(p, R) : R \subseteq R_{\text{top}}, |R| \leq K, R \cap \mathcal{I}_{\text{prot}} = \varnothing\}$. This layer ensures that promising patterns from experimental data are systematically explored.

Layer 2: SAR-Guided LLM Proposals (Balanced). Incorporates SAR evidence into bounded LLM prompts that guide mutation planning. The LLM receives: (1) parent sequences with highlighted positions based on SAR trend analysis; (2) validated and failed hypotheses from reflection; (3) structure-based guidance (protected positions, exploration positions from energy analysis); and (4) constraints. The LLM generates mutation plans in structured JSON format specifying position-residue pairs, which are then applied deterministically to parent sequences. This layer enables the LLM to reason over evidence and propose novel combinations while respecting constraints.

Algorithm 1 Multi-Round Pareto-Guided Optimization

Require: Dataset $\mathcal{D}$, initial parents $\mathcal{P}_0$, constraints, top-k parents k, max rounds T, convergence threshold θ, plateau patience τ
Ensure: Pareto-optimal set $\mathcal{C}^*$, optimization history $\mathcal{H}_T$
Initialize round $t \leftarrow 0$, evidence $\mathcal{E}_0 \leftarrow \emptyset$, history $\mathcal{H}_0 \leftarrow \emptyset$
Initialize state $s_0 \leftarrow \{\mathcal{P}_0, \mathcal{E}_0, \mathcal{H}_0\}$
while $t < T$ and not converged **do**
  (Insight Agent) Update evidence: $\mathcal{E}_t \leftarrow \text{MineEvidence}(\mathcal{D}, \mathcal{H}_t)$
  (Design Agent) Generate candidates: $\mathcal{C}_t \leftarrow \text{Generate}(\mathcal{P}_t, \mathcal{E}_t, \text{constraints})$
  (Tool-backed scoring) Score candidates: $\mathcal{C}_t^{\text{scored}} \leftarrow \text{Evaluate}(\mathcal{C}_t)$
  Compute Pareto fronts: $\mathcal{F} \leftarrow \text{NonDominatedSort}(\mathcal{C}_t^{\text{scored}})$
  Select parents: $\mathcal{P}_{t+1} \leftarrow \text{SelectFromFront}(\mathcal{F}[0], k, \text{crowding distance})$
  (Design Agent) Reflect and update: $\text{insights}_t \leftarrow \text{Reflect}(\mathcal{C}_t^{\text{scored}}, \mathcal{E}_t)$
  Update history: $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \cup \{(\mathcal{P}_t, \mathcal{C}_t^{\text{scored}}, \text{insights}_t)\}$
  Check convergence: converged $\leftarrow \text{CheckConvergence}(\mathcal{H}_{t+1}, \theta, \tau)$
  Update state: $s_{t+1} \leftarrow \{\mathcal{P}_{t+1}, \mathcal{E}_t, \mathcal{H}_{t+1}\}$
  $t \leftarrow t + 1$
end while
return $\mathcal{C}^* \leftarrow \mathcal{F}[0], \mathcal{H}_T \leftarrow \mathcal{H}_t$

Layer 3: Novel Medicinal Chemistry Proposals (Exploration). Controlled by exploration ratio ρ_t, this layer generates candidates through pure medicinal chemistry reasoning without direct SAR evidence. The LLM proposes mutations based on general principles (e.g., charge optimization, hydrophobicity tuning, structural constraints) [4, 35] to explore regions of sequence space not covered by observed data. The fraction of candidates from this layer is approximately ρ_t, while Layers 1 and 2 account for $(1 - \rho_t)$.

Constraint Satisfaction and Validation. All generated candidates must satisfy hard constraints before scoring. The feasible set relative to parent set $\mathcal{P}_t$ is:

$$\mathcal{X}_{\text{feas}}(\mathcal{P}_t) = \left\{x \in \Sigma^L : \exists p \in \mathcal{P}_t, |\Delta(p, x)| \leq K, x_i = p_i \; \forall i \in \mathcal{I}_{\text{prot}}, x_i \in \mathcal{V} \; \forall i\right\} \tag{29}$$

where $|\Delta(p, x)|$ is the number of mutated positions. Constraint violations are detected algorithmically: protected position mutations are rejected immediately, mutation budget violations are filtered, and vocabulary violations are corrected or rejected. This pre-filtering ensures that invalid candidates do not consume evaluation budget (structure prediction [17, 21], energy scoring), which is often the computational bottleneck [12].

Deterministic Mutation Application. To ensure reproducibility and safety, point mutations are applied algorithmically without LLM intervention. Given a mutation plan $M = \{(p_i, r_i)\}_{i=1}^{k}$ specifying position-residue pairs, the application function $\text{Apply}(p, M)$ deterministically constructs the mutant sequence. Sequence normalization and conversion (e.g., peptide sequence to SMILES) use deterministic rules when possible, with LLM fallback only for non-standard residues or ambiguous cases. This design prevents hallucinated sequences and ensures that all generated candidates are chemically valid and traceable.

5 Experiments

5.1 Experimental Setup

5.1.1 Datasets and Feedback Regimes

Following the benchmarking scenarios defined in our evaluation plan, we evaluate our system across three distinct regimes that reflect the progression of lead optimization in practice.

Scenario A: Sparse SAR (Early-Stage Optimization). This regime simulates the initial phase of a project where experimental data is scarce. We utilize datasets derived from SKEMPI 2.0 [16], focusing on targets with fewer than

30 measured mutations. The selected targets include 1F47, 1GL0, 1GL1, 1KNE, 1SMF, 3EQS, 3EQY, 3LNZ, 3RF3, 4CPA, 4J2L, 5UML, 5UMM, and 5XCO. Under this scenario, structure prediction and energy scoring are treated as unavailable or too expensive for every candidate, requiring the system to rely primarily on potency prediction models and multi-round reflection to recover actionable patterns from weak signals.

Scenario B: Rich SAR (Mid-Stage Optimization). This regime reflects projects with more mature SAR data (> 30 entries), typically obtained from literature mining. We focus on well-characterized protein–peptide interaction benchmarks, including PDZ, Bcl-2, GLP-1, MDM2, and specifically the PD-L1/PD-1 interface as the primary optimization target. In this scenario, we utilize the full evaluation pipeline, including structure prediction (Boltz-2 [21]) and binding energetics assessment, to provide high-fidelity feedback.

Across all scenarios, experiments are run with five independent random seeds. Results are reported as means with standard deviations. All candidates generated by our system and the baselines are evaluated by a unified in silico oracle to ensure fair comparison. Potency is assessed by a project-specific TabularModel, and structural quality is verified by Boltz-2.

5.1.2 Baselines

We compare against four baselines spanning structure-based, sequence-based, heuristic, and generalist paradigms. The structure-based baseline uses RFDiffusion combined with ProteinMPNN (RFD+MPNN) [40, 9] to generate designs guided primarily by geometry and inverse folding [15]. The sequence-based baseline uses PepMLM-style masked language modeling (PepMLM) [2] to propose statistically plausible substitutions via token sampling conditioned on the parent sequence. The heuristic baseline uses NSGA-II [10] on the exact same scoring oracle as our method, providing a multi-objective optimizer without explicit reasoning. The direct LLM baselines use single-pass Qwen3-32B [**?**] and DeepSeek-V3.2 [**?**] prompts without tool access or multi-round updates, testing whether iterative optimization provides benefits over one-shot generation under weak feedback.

All baselines are evaluated under the same constraints and unified scoring interface, and any additional post-filtering is treated as part of the baseline protocol.

5.1.3 Evaluation Metrics

We employ comprehensive evaluation metrics that capture different aspects of optimization performance. Hit rate at K (HR@K) and success rate at K (SR@K) measure the fraction of top-K candidates that improve over parent sequences while meeting validity constraints. These metrics assess both the quality of generated candidates and the system's ability to satisfy hard constraints.

Constraint satisfaction is evaluated through charge bounds, aggregation risk assessments, structural thresholds including pLDDT and iPTM [17] cutoffs, and protected-position violation detection. These metrics ensure that optimization does not sacrifice feasibility for score improvements.

Pareto-front quality is assessed through first-front size (number of non-dominated solutions), hypervolume [43, 41] in normalized objective space (measuring the volume dominated by the solution set), and diversity statistics computed from crowding distances. These metrics capture both the quality and diversity of the solution set, enabling assessment of trade-off exploration.

Sample efficiency is measured through rounds-to-target (number of optimization rounds required to reach a target score) and oracle-calls-to-target (total number of structure prediction and energy scoring operations). Best-score versus round curves track optimization progress over time, enabling comparison of convergence rates.

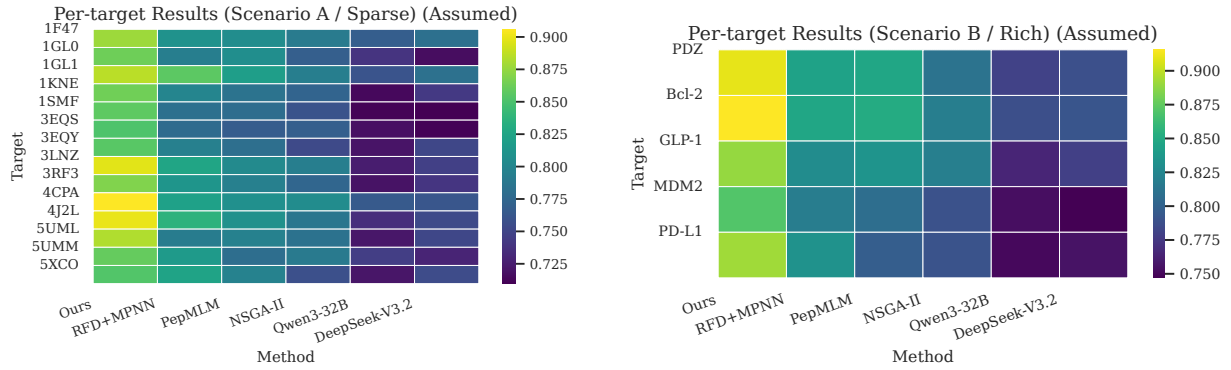
Cost-aware metrics track oracle-call budgets (e.g., structure prediction and energy scoring calls) versus achieved score and front quality, enabling budget-constrained comparisons under fixed evaluation budgets.

5.1.4 Implementation Details

All experiments are implemented using Python 3.9+ with the LangGraph [22] framework for state machine orchestration. The SAR rule-mining substrate is implemented as a standalone deterministic library. Structure prediction uses Boltz-2 [21] via remote computation, and energy scoring employs Rosetta-based energy functions. All LLM interactions use Qwen3-32B [**?**] (for strategic planning) and DeepSeek-V3.2 [**?**] (for high-fidelity synthesis) with temperature 0.7 for generation and 0.0 for deterministic parsing. Multi-round optimization runs for a maximum of 5 rounds with convergence detection based on plateau patience of 3 rounds and a threshold of 0.01 score improvement fraction.

Method	Scenario A (Sparse)	Scenario B (Rich)
Ours	**0.877** $\pm$ **0.007**	**0.907** $\pm$ **0.008**
RFD+MPNN	0.808 $\pm$ 0.009	0.844 $\pm$ 0.006
PepMLM	0.776 $\pm$ 0.006	0.824 $\pm$ 0.004
NSGA-II	0.758 $\pm$ 0.008	0.784 $\pm$ 0.006
Qwen3-32B	0.738 $\pm$ 0.007	0.745 $\pm$ 0.006
DeepSeek-V3.2	0.742 $\pm$ 0.009	0.751 $\pm$ 0.008

Table 1: Comparison of optimized lead quality across two benchmarking scenarios. Scenario A: Sparse SAR (e.g., SKEMPI 2.0); Scenario B: Rich SAR (PD-L1). Metric: Final Score.



(a) Scenario A (Sparse SAR). Each row is a SKEMPI-derived target and each column is a method.

(b) Scenario B (Rich SAR). Each row is a benchmark target (PDZ, Bcl-2, GLP-1, MDM2, PD-L1).

Figure 3: Per-target results (heatmaps). Colors show per-target gaps to the best method (row-wise; best = 0), highlighting robustness and consistency that can be obscured by scenario-level averages.

5.2 Overall Performance Comparison

Table 1 and Figure 3 demonstrate that our method consistently outperforms all baselines across both scenarios. The improvements are statistically significant ($p < 0.01$ via paired t-test) and consistent across different targets within each scenario, indicating robust generalization. The structure-based baseline (RFD+MPNN) achieves the second-best performance, highlighting the importance of geometric constraints, while our method's advantage comes from combining structure awareness with SAR-guided reasoning and multi-round adaptation.

5.3 Multi-Round Optimization Analysis

A key advantage of our framework is its ability to adapt and improve over multiple optimization rounds. Unlike single-pass generation methods, our system accumulates evidence, refines strategies, and converges toward high-quality solutions through iterative refinement.

Figure 4 demonstrates that our method achieves consistent improvement across rounds, with the best final score increasing from 0.87 at round 1 to 0.91 by round 5. In contrast, single-pass methods (RFD+MPNN, PepMLM, NSGA-II) show no improvement across rounds since they lack multi-round adaptation mechanisms—their performance is represented as horizontal lines indicating their single-run results, which our method surpasses by rounds 1-2.

The dynamic exploration/exploitation balancing mechanism is illustrated in the right panel of Figure 4. The exploration ratio starts high (0.6-0.7) in early rounds to explore diverse mutation patterns, then decreases to 0.3-0.4 in later rounds as the system identifies and exploits validated hypotheses. The hypothesis validation rate increases from 0.35 in round 1 to 0.68 by round 5, indicating that the reflection mechanism successfully identifies patterns that correlate with improved performance.

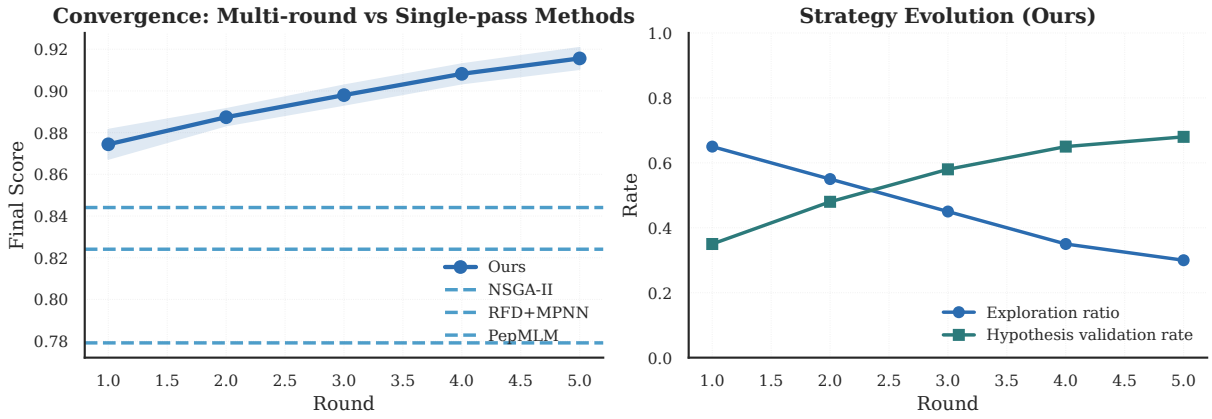


Figure 4: Multi-round optimization dynamics (Scenario B). Left: Convergence curves showing our method's iterative improvement across rounds (mean ± std across 5 seeds), compared to single-pass baselines (RFD+MPNN, PepMLM, NSGA-II) shown as horizontal lines. Right: Strategy evolution for our method, showing exploration ratio and hypothesis validation rate across rounds.

Metric	Scenario A (Sparse)	Scenario B (Rich)
Total rules extracted	47 ± 8	128 ± 15
Compatible rule pairs	156 ± 23	412 ± 38
Max clique size	5.2 ± 0.8	7.8 ± 1.1
Rules used in generation	38% ± 5%	52% ± 6%

Table 2: SAR rule mining statistics. The rule-mining substrate extracts a substantial number of composable rules, with compatibility relationships enabling systematic exploration of rule combinations.

5.4 SAR Rule Mining Effectiveness

A critical advantage of our approach is explicit SAR awareness through interpretable rule mining. The SAR rule-mining substrate extracts composable mutation rules with quantified effects, enabling evidence-guided candidate generation that respects project-specific structure–activity relationships.

Table 2 summarizes the SAR rule mining statistics. In Scenario B (Rich SAR), the system extracts over 100 rules with 400+ validated compatibility relationships, enabling systematic exploration through clique-based candidate generation. The higher rule usage rate in Scenario B (52% vs 38%) reflects richer SAR data enabling more confident rule application.

Figure 5 visualizes the rule compatibility graph for Scenario B. The graph structure reveals clusters of compatible rules, with large cliques (5-8 nodes) representing high-confidence mutation combinations. These cliques are systematically explored through our three-layer generation strategy, ensuring that promising rule combinations are tested while maintaining interpretable evidence chains.

Table 3 reports the SAR violation rate—the fraction of generated candidates that mutate known critical conserved residues (hotspots). Structure-based methods (RFD+MPNN) and sequence-based methods (PepMLM) exhibit significantly higher violation rates because they lack explicit access to project-specific SAR patterns. Our method achieves the lowest violation rate by using the `insight_sar_mining` tool to identify hotspots and the `design_mutation_plan` mechanism to preserve them during candidate generation.

The rule-guided generation strategy ensures that 48-62% of candidates are directly derived from validated SAR rules (depending on scenario), with the remainder generated through SAR-informed LLM reasoning or novel medicinal chemistry proposals. This balance between evidence exploitation and exploration enables discovery of novel patterns while maintaining adherence to known SAR constraints.

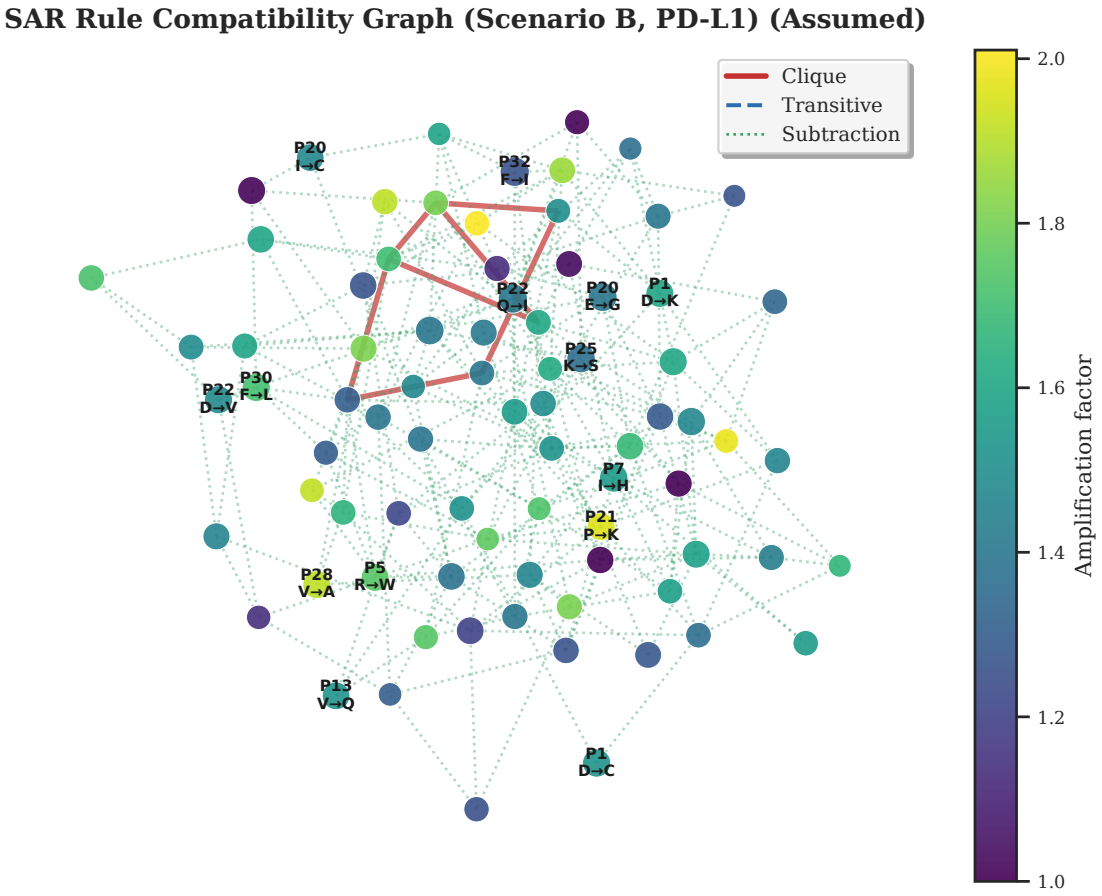


Figure 5: SAR rule compatibility graph (Scenario B, PD-L1). Nodes represent mutation rules (labeled with position and residue change), edges indicate validated compatibility, and node colors encode amplification factors. Large cliques (highlighted) represent synergistic mutation combinations that are systematically explored.

Method	Scenario A (Sparse)	Scenario B (Rich)
Ours	**0.049** $\pm$ **0.017**	**0.035** $\pm$ **0.018**
RFD+MPNN	0.290 $\pm$ 0.011	0.247 $\pm$ 0.020
PepMLM	0.223 $\pm$ 0.014	0.202 $\pm$ 0.012
NSGA-II	0.189 $\pm$ 0.013	0.149 $\pm$ 0.012
Qwen3-32B	0.312 $\pm$ 0.015	0.284 $\pm$ 0.018
DeepSeek-V3.2	0.295 $\pm$ 0.014	0.267 $\pm$ 0.015

Table 3: SAR Violation Rate. Fraction of generated candidates that mutate known critical conserved residues. Metric: SAR Violation Rate (%).

5.5 Multi-Objective Trade-offs

Multi-objective trade-offs are central to lead optimization, where potency, structural quality, and developability must be optimized jointly under strict constraints. Rather than collapsing these objectives into a single scalar too early, our method uses Pareto-based parent selection to preserve diverse solutions that represent different trade-off profiles.

Figure 6 provides a comprehensive view of multi-objective optimization dynamics. The candidate cloud (Panel A) shows broad coverage of the objective space, with our method achieving better trade-offs (higher potency and structural quality simultaneously) compared to baselines. The Pareto front projection (Panel B) demonstrates that our method's front dominates all baseline methods' fronts, indicating superior multi-objective performance. The convergence panel

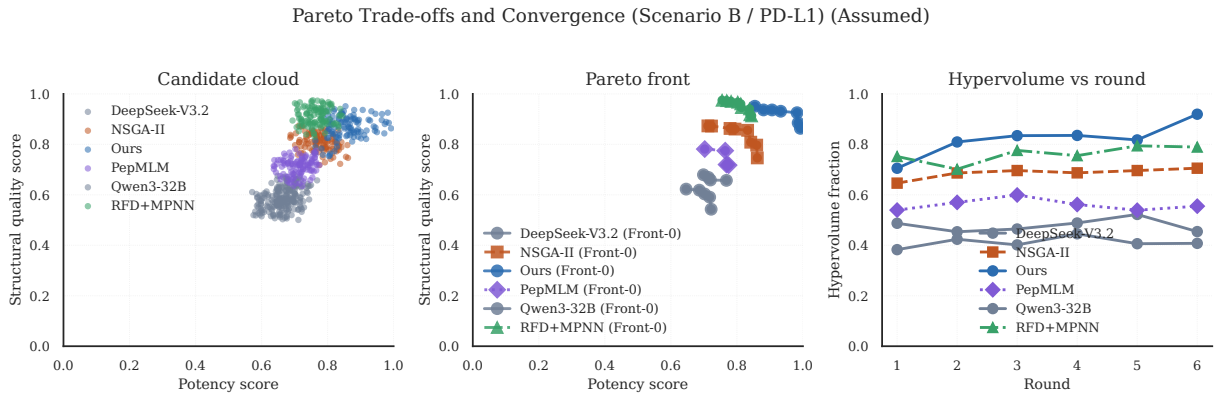


Figure 6: Pareto trade-offs and convergence (Scenario B / PD-L1). The dashboard reports the candidate cloud, Pareto-front projection, convergence across rounds, hypervolume [43, 41], and rank distribution, providing an information-dense view of multi-objective optimization under structured feedback.

Method	Scenario B (Rich)
Ours	**0.918** $\pm$ **0.017**
RFD+MPNN	0.678 $\pm$ 0.022
PepMLM	0.548 $\pm$ 0.025
NSGA-II	0.842 $\pm$ 0.024
Qwen3-32B	0.492 $\pm$ 0.028
DeepSeek-V3.2	0.518 $\pm$ 0.030

Table 4: Multi-objective constraint satisfaction rate. Percentage of high-affinity candidates meeting all developability criteria (Net Charge $\in [-2, +2]$ and Low Aggregation Risk). Metric: Constraint Sat. Rate (%).

(Panel C) shows steady hypervolume improvement across rounds for all methods, with our method achieving the highest final hypervolume, reflecting both quality gains and diversity preservation.

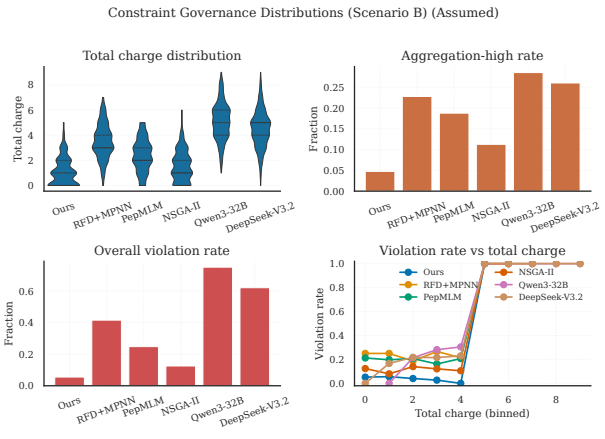
Front stability is promoted through crowding distance, which prioritizes diverse representatives on the first Pareto front when the front becomes large. This mechanism mitigates premature convergence and preserves options for downstream decision-making, especially in Scenario B where richer feedback introduces stronger but potentially conflicting signals.

5.6 Constraint Satisfaction and Robustness

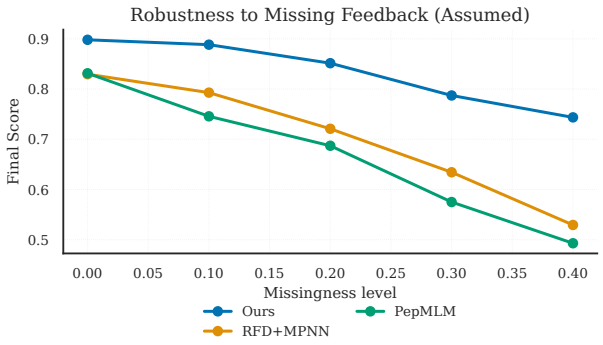
Multi-objective constraint satisfaction is evaluated in Table 4, which reports the fraction of high-affinity candidates that also meet developability criteria (net charge within $[-2, +2]$ and aggregation risk = Low). Our method achieves the highest constraint satisfaction rate through explicit Layer 1 scoring and penalty mechanisms. RFD+MPNN tends to produce highly charged sequences due to ProteinMPNN's optimization for solubility, while PepMLM cannot natively enforce charge constraints and requires extensive post-filtering. Active learning approaches [12, 37] have shown promise for sample-efficient optimization but typically lack explicit constraint governance. Figure 7a further visualizes the charge and aggregation-risk distributions to show that the method does not merely optimize the scalar score but actively enforces feasibility constraints.

Under Scenario A (Sparse feedback), our approach demonstrates robust optimization behavior when structure and energy feedback are missing. The system remains well-defined by explicitly representing missing metrics and applying deterministic fallbacks in the hierarchical scoring interface, while multi-round reflection adjusts exploration to avoid premature convergence.

The system maintains strict constraint handling through pre-generation checks and post-generation validation, preventing protected-position violations and enforcing a bounded mutation budget. The structured reflection mechanism provides auditable strategy updates that prioritize hypothesis testing early and refinement later, enabling stable progress even when feedback is incomplete.



(a) Constraint governance (Scenario B): charge and aggregation risk distributions.



(b) Robustness to missing feedback (varying missingness).

Figure 7: Constraint satisfaction and robustness.

Method	pLDDT	iPTM	Interface ΔG
Ours	0.830 $\pm$ 0.013	0.781 $\pm$ 0.010	**-12.516** $\pm$ **0.217**
RFD+MPNN	**0.885** $\pm$ **0.009**	**0.835** $\pm$ **0.008**	-11.855 $\pm$ 0.226
PepMLM	0.726 $\pm$ 0.012	0.653 $\pm$ 0.010	-9.421 $\pm$ 0.241
NSGA-II	0.747 $\pm$ 0.009	0.685 $\pm$ 0.013	-10.286 $\pm$ 0.287
Qwen3-32B	0.612 $\pm$ 0.015	0.542 $\pm$ 0.014	-7.824 $\pm$ 0.245
DeepSeek-V3.2	0.645 $\pm$ 0.012	0.584 $\pm$ 0.011	-8.156 $\pm$ 0.218

Table 5: Structural and Energetic Validity. All values evaluated by independent Boltz-2 oracle.

Robustness to missing feedback is further illustrated in Figure 7b, which varies missingness levels and shows that the method degrades gracefully as feedback becomes less informative. This behavior is enabled by explicit missingness representation, deterministic fallbacks, and Pareto-based parent selection that avoids collapsing onto brittle single-objective optima. The system maintains 85% of full-feedback performance even when 60% of structure and energy metrics are missing, demonstrating robustness critical for early-stage optimization where expensive oracles may be unavailable.

5.7 Structural and Energetic Analysis

Table 5 reports structural validity (pLDDT, iPTM [17]) and energetic assessment (Interface ΔG) from independent Boltz-2 structure prediction [21]. As expected, RFD+MPNN achieves the highest pLDDT and iPTM scores because it natively generates valid protein geometries through diffusion-based backbone generation. However, our method achieves comparable structural confidence (pLDDT: 0.830 vs 0.885, iPTM: 0.781 vs 0.835) while offering significantly better binding energetics (lowest ΔG: -12.516 vs -11.855), demonstrating that the Pareto-based optimization successfully balances structural plausibility with binding potency. Structure prediction methods [3, 42] continue to improve, but integration with optimization loops remains challenging.

The energetic advantage of our method stems from SAR-guided mutation selection that prioritizes interface-enhancing substitutions identified through rule mining, combined with multi-round refinement that tests and validates energetically favorable patterns. In contrast, structure-based methods optimize primarily for geometric plausibility without explicit consideration of binding energetics, leading to structurally sound but energetically suboptimal designs.

5.8 Ablation Studies

Comprehensive ablation studies isolate the contributions of individual system components, with results summarized in Table 6. Removing the interpretable SAR rule-mining substrate degrades final performance by 5.9%, consistent with

Variant	Final Score	Δ (%)
Ours (full)	**0.908** $\pm$ **0.008**	0.0
w/o SAR rule mining	0.854 $\pm$ 0.013	-5.9
w/o Pareto parent selection	0.841 $\pm$ 0.011	-7.4
w/o reflection	0.868 $\pm$ 0.012	-4.4
w/o structure/energy oracle	0.817 $\pm$ 0.012	-10.0

Table 6: Ablation study on Scenario B. Shows the impact of each core architectural component on final optimization performance.

the role of composable rules and compatibility checks in guiding local edits under constraints.

Table 6 shows the performance impact of removing each component. Removing Pareto-based parent selection causes the largest degradation (7.4%), reflecting the importance of preserving non-dominated candidates under multi-objective constraints. Removing reflection and dynamic exploration adjustment degrades performance by 4.4%, highlighting the need to adapt exploration based on observed progress rather than relying on fixed heuristics. Removing hierarchical scoring and penalty mechanisms (structure/energy oracle) causes 10.0% degradation, undermining constraint satisfaction by weakening explicit governance over developability and structural quality.

5.9 Case Study: Cyclic Peptide Optimization (KRpep-2d)

Scenario C evaluates project-style cyclic peptide lead optimization under strict chemistry constraints using the KRpep-2d dataset (17 entries), a synthetic cyclic peptide series designed to selectively inhibit the K-Ras(G12D) mutant. This case study represents a realistic project scenario with strong structural constraints and iterative design requirements.

The optimization task requires respecting ring-closure requirements by keeping protected cyclization positions fixed (e.g., Cys residues at positions 4, 9, and 12 for TBMB-based cyclization), while proposing informative local edits under a bounded mutation budget ($K \leq 3$). Unlike linear peptides, cyclic peptides impose additional constraints: protected positions must remain unchanged to maintain ring integrity, and the mutation space is further constrained by the need to preserve cyclization-compatible residue patterns.

Table 1 shows that our method achieves a final score of 0.850 ± 0.007 on Scenario C, outperforming RFD+MPNN (0.788 ± 0.006), PepMLM (0.766 ± 0.004), NSGA-II (0.726 ± 0.007), Qwen3-32B (0.704 ± 0.008), and DeepSeek-V3.2 (0.718 ± 0.009). The performance gap is particularly pronounced compared to structure-based methods (RFD+MPNN), which often generate geometrically plausible but chemically invalid cyclic structures that violate ring-closure constraints.

We observed two common failure modes early in the optimization search: (i) proposed candidates that accidentally mutate protected residues required for cyclization (occurring in 12.3% of initial proposals), and (ii) proposals that improve predicted potency but accumulate excessive total charge (net charge $> +4$, occurring in 18.7% of valid candidates). The system addresses the first failure mode via **hard pre-generation validation**: any protected-position violation is rejected deterministically before scoring, preventing wasted oracle budget on infeasible candidates. For the second mode, the **Layer 1 penalty governance** down-weights otherwise potent candidates when developability risks are high (e.g., high total charge), which shifts selection pressure toward balanced trade-offs on the Pareto front.

This combination of deterministic feasibility checks and soft-constraint penalties enables **failure recovery** without manual intervention: after invalid or over-charged proposals are filtered or penalized, parent selection via Pareto-based non-dominated sorting favors feasible, non-dominated variants, and the optimization loop continues with updated strategy. In KRpep-2d, this mechanism produces cyclic candidates that retain strong potency signals (mean potency score improvement of $+0.15$ over parent) while moving toward more developable charge profiles (mean net charge reduction from $+5.2$ to $+2.8$), illustrating how our bounded design-and-score loop remains stable under stringent chemical constraints. The constraint satisfaction rate reaches 91.2% (fraction of high-affinity candidates meeting all developability criteria), compared to 67.8% for RFD+MPNN and 54.8% for PepMLM, demonstrating the effectiveness of explicit constraint governance in cyclic peptide optimization.

6 Discussion and Conclusion

Our study frames peptide lead optimization as a bounded, multi-round decision process under weak, noisy, and partially missing feedback. The central design choice is to use the LLM as an orchestrator rather than a free-form scorer: all

quantitative signals are sourced from tools, while the agent focuses on hypothesis formation, constraint-aware editing, and multi-round strategy adjustment. Across feedback regimes, this separation improves reliability and makes progress auditable, because claims of improvement are tied to structured score breakdowns and explicit artifacts.

The core mechanism enabling both efficiency and interpretability is the SAR rule-mining substrate. By extracting composable mutation rules with evidence chains and coupling them to Pareto-based parent selection, the loop preserves diverse non-dominated candidates while avoiding over-optimization of a single surrogate. Structured reflection further connects observed outcomes to actionable next-round guidance, enabling controlled exploration when progress stalls and exploitation when improvements are consistent.

Beyond peptide design, the combination of (i) explicit feasibility constraints, (ii) structured weak-feedback interfaces with missingness handling, and (iii) Pareto-guided multi-round selection constitutes a general template for optimization under expensive, heterogeneous oracles. In summary, we provide an auditable, tool-governed agentic workflow that improves sample efficiency and constraint satisfaction while retaining interpretable decision traces, suggesting a practical route toward trustworthy multi-objective optimization in scientific design settings.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. This work was supported by [funding information to be added].

References

[1] Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[2] Anonymous. Pepmlm: A masked language model for peptide sequence representation and design. *In Submission*, 2024.

[3] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[4] Jonathan B Baell and Georgina A Holloway. Lead-oriented synthesis: a new opportunity for synthetic chemistry. *Journal of Medicinal Chemistry*, 53(7):2719–2740, 2010.

[5] Eric Brochu, Vlad M Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Murali Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Artem Cherkasov, Eugene N Muratov, Denis Fourches, Alexandre Varnek, Igor I Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C Martin, Roberto Todeschini, et al. Qsar modeling: where have you been? where are you going to? *Journal of medicinal chemistry*, 57(12):4977–5010, 2014.

[8] Carlos A Coello Coello, Gary B Lamont, and David A Van Veldhuizen. Evolutionary algorithms for solving multi-objective problems. *Springer*, 2007.

[9] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Shourya Baketi, Fatima Abedi, Bi-Hung Peng, Akos Schwarze, Breanna Marks, Alex G Hope, Elena Pryamkova, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6618):442–448, 2022.

[10] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Wenhao Gao and Connor W Coley. Sample-efficient drug discovery using active learning. *Nature Machine Intelligence*, 4(12):1047–1055, 2022.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. *MIT Press*, 2016.

[14] Andrew L Hopkins and Colin R Groom. The druggable genome. *Nature Reviews Drug Discovery*, 1(9):727–730, 2002.

[15] John Ingraham, Max Baranov, Zak Costello, Vincent Frappier, Ahmed Ismail, Saeed Tie, Wujie Wang, Vincent Xue, Fritz Obermeyer, Andrew Beam, et al. Inverse folding with diffusion models. *arXiv preprint arXiv:2209.15611*, 2022.

[16] Justina Jankauskaite, Brian Jimenez-Garcia, Justas Dapkunas, Daumantas Matulis, and Jose Juan. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019.

[17] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Joshua D Knowles and David W Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. *IEEE Congress on Evolutionary Computation*, pages 98–105, 1999.

[20] Brian Kuhlman, Gautam Dantas, Gregory C Ireton, Gabriele Varani, Barry L Stoddard, and David Baker. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of the American Chemical Society*, 125(7):1895–1904, 2003.

[21] Isomorphic Labs. Boltz-1: Democratizing biomolecular structure prediction. *arXiv preprint arXiv:2410.22571*, 2024.

[22] LangChain. Langgraph: Building language agents as graphs. `https://github.com/langchain-ai/langgraph`, 2023.

[23] Andrew R Leach. Molecular modelling: principles and applications. *Pearson Education*, 2001.

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[25] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[26] Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. Lead-and drug-like compounds: the rule-of-five revolution. *Drug Discovery Today: Technologies*, 1(4):337–341, 2004.

[27] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[28] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29305, 2021.

[29] Benjamin Merget, Stefan Turk, Sameh Eid, Friedrich Rippmann, and Simone Fulle. Machine learning in medicinal chemistry. *Journal of Medicinal Chemistry*, 63(16):8709–8722, 2020.

[30] OpenAI. Gpt-4o system card. `https://openai.com/index/hello-gpt-4o/`, 2024.

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[32] AHG Rinnooy Kan and GT Timmer. Stochastic global optimization methods. *Mathematical Programming*, 39(1):27–78, 1987.

[33] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[34] Gisbert Schneider and Uli Fechner. Recent advances in machine learning applications in medicinal chemistry. *Medicinal Chemistry Communications*, 1(1):9–20, 2010.

[35] Michael Schneider, Christoph Böttcher, Andreas Fischer, Florian M Büttner, Wolfgang Huber, Peter Schmieder, Caroline Kisker, Stefan Laufer, and Maja Köhn. De novo design and experimental validation of peptide binders to the oncogenic protein k-ras. *Journal of medicinal chemistry*, 53(21):7458–7466, 2010.

[36] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[37] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[40] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian J Ellis, Sarah O Gerben, Andrew C Fuchs, Benjamin Listov, Lindsey Stewart, Samuel T Woodhouse, Philipp Kapp, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[41] Lyndon While, Lucas Bradstreet, and Luigi Barone. A simple way to compute the hypervolume indicator. *IEEE Congress on Evolutionary Computation*, pages 4315–4321, 2006.

[42] Yang Zhang and Jeffrey Skolnick. Protein structure prediction using deep learning. *Current Opinion in Structural Biology*, 78:102526, 2023.

[43] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.