AMP-MODIFY: CHEMICAL MODIFICATION DESIGN OF ANTIMICROBIAL PEPTIDES WITH A CLOSED-LOOP FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Antimicrobial peptides (AMPs) offer a promising route to combating drug-resistant infections, but their therapeutic development requires addressing limitations in stability, selectivity, and activity. Chemical modifications, such as lipidation, cyclization, and terminal capping, provide a critical way to tune these properties beyond amino acid sequence optimization. However, computational AMP design has largely focused on sequence generation, leaving chemical modification underexplored. We formulate the modification problem as a joint decision over *whether* to modify a parent peptide, *where*, and *what* chemical structure to introduce. A central challenge is the mismatch between abundant general chemical knowledge and scarce experimental data linking parent peptides, modifications, and outcomes. Here, we develop AMP-Modify to address it by separating chemically explicit construction and validation from AMP-specific effect prediction. An LLM-assisted search uses general chemical knowledge to propose edit strategies within a modification space defined by a rule registry, while a molecular compiler instantiates and validates candidates. An AMP-specific modification evaluator trained on $8{,}145$ parent-modified relations scores each candidate relative to its parent, and its predictions are fed back to guide subsequent search and final selection. The evaluator outperforms existing methods in ranking modification effects, while case studies show that modifications proposed by our framework align with experimentally observed trends. AMP-Modify extends computational AMP design beyond sequence generation to chemically explicit modification.

1 INTRODUCTION

Antimicrobial peptides (AMPs) are an important class of candidates for new anti-infective agents (Hancock & Sahl, 2006; Mahlapuu et al., 2016). They often act through mechanisms that differ from those of conventional antibiotics, which makes them attractive against drug-resistant pathogens. Natural AMPs, however, rarely meet the combined needs of potency, selectivity, and stability. Chemical modifications, including terminal capping, lipidation, cyclization, and non-natural residue substitution, provide ways to tune these properties beyond sequence optimization (Huan et al., 2020). This motivates a distinct design problem: given an existing AMP and a target objective, decide *whether* to modify it, *where* to make the change, and *what* chemical structure to introduce.

This problem differs from sequence generation in both its representation and its objective. A residue string alone cannot fully specify attachment sites, lipid chain lengths, ring connectivity, or stereochemistry. Chemically distinct variants may therefore share the same sequence while exhibiting different properties. Moreover, predicting that a modified peptide is active does not establish that the modification benefits its parent. An already potent parent may remain active after a detrimental edit. Modification design thus requires chemically explicit candidates and a parent-relative assessment: does this particular change improve the starting peptide against the target organism group? Retaining the parent is part of the decision, not a failure to generate a candidate.

Existing methods address parts of this problem but do not bring these requirements together. Most computational AMP design methods generate or optimize amino-acid sequences (Szymczak et al.,

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.

2023; Van Oort et al., 2021; Zhou et al., 2026). Approaches that incorporate non-canonical chemistry either restrict the editable space to terminal groups (Xu et al., 2026) or cover only a small set of modification types and encode modifications as discrete tokens (Jung et al., 2026). Modification-aware predictors can assess a specified edit, but they do not design or construct candidates (Agrawal & Raghava, 2018; Kumar et al., 2020). These are useful extensions, but their supported choices and structural detail remain limited. The remaining challenge is to connect chemically explicit construction, parent-relative evaluation, and candidate selection within a cohesive design process.

The main obstacle is a mismatch between the breadth of the design space and the available supervision. Learning a broad modification generator directly would require examples spanning modification types, sites, structural parameters, and peptide contexts. Experimentally measured parent–modified pairs cover only a small part of this space. Yet not every part of the problem requires such data. General chemical knowledge can suggest which transformations to explore. Explicit rules can determine applicable sites and support structural validation. Paired experimental outcomes are most directly needed to learn how a specified modification affects a particular parent. This distinction motivates separating proposal, construction, and effect prediction, rather than asking a single model to learn all three from scarce paired examples.

We implement this decomposition and develop **AMP-Modify**. To make this exploration chemically explicit and verifiable, a rule registry defines a modification space, comprising 20 construction types across five categories. An LLM-assisted search uses general chemical knowledge and accumulated observations to propose edit strategies. A deterministic compiler instantiates these proposals, returns molecular structures and structured events, and rejects candidates that fail the chemical constraints. A modification-effect evaluator (AMP-Modify-Eval) then combines sequence, molecular, and event representations to score each candidate relative to its parent, alongside absolute activity and toxicity outputs. These scores guide the next search step and final selection. The LLM therefore proposes edits in a registered space; it does not decide chemical validity or parent-relative effect. The compiler and the evaluator provide those, respectively. The procedure returns up to K qualified candidates, or an empty shortlist when none meets the requested pair-direction criterion (Figure 1).

Our experiments test three aspects of this formulation: whether parent-relative supervision is needed for effect prediction, whether explicit construction makes LLM-assisted exploration more effective than LLM-only proposals, and whether evaluator feedback improves subsequent search. Using a corpus of $8{,}145$ parent–modified relations, we evaluate ranking on 980 high-confidence pairs under connected-component splits. AMP-Modify-Eval achieves a family-balanced Spearman correlation of 0.378, compared with 0.281 for the strongest baseline on this metric; removing the parent-relative objective reduces this correlation to 0.053. In end-to-end search under matched proposal budgets, AMP-Modify converts 89.7% of proposal slots into unique-valid candidates, compared with 38.0% for a Raw LLM that uses the same model without the registry or compiler, and reaches structurally specific published edits such as an Ahx substitution and a glutamate side-chain cyclization that the ungrounded proposals do not produce. Withholding evaluator feedback while retaining the same structured search reduces evaluator-qualified improvement yield from 57.3% to 48.4%.

Our main contributions are:

- We formulate AMP modification as a chemically explicit, parent-relative design problem that jointly addresses whether to modify, where, and what structure to introduce.
- We develop AMP-Modify, which separates rule-based construction from parent-relative effect prediction and uses the resulting scores to guide LLM-assisted search and selection.
- We introduce AMP-Modify-Eval and evaluate both its parent-relative predictions and its role in search through held-out benchmarks, literature cases, and controlled ablations.

2 Related Work

Sequence-level AMP design. Most computational methods for AMP design focus on generating or optimizing amino-acid sequences (Szymczak et al., 2023; Van Oort et al., 2021). Recent approaches adopt generative models, language models, or multi-objective optimization to search for peptides with desired potency, safety, and stability. MAC-AMP (Zhou et al., 2026), for example, treats AMP design as a multi-objective generation problem and iteratively refines candidate sequences. While these methods have advanced controllable AMP generation, their sequence-centric

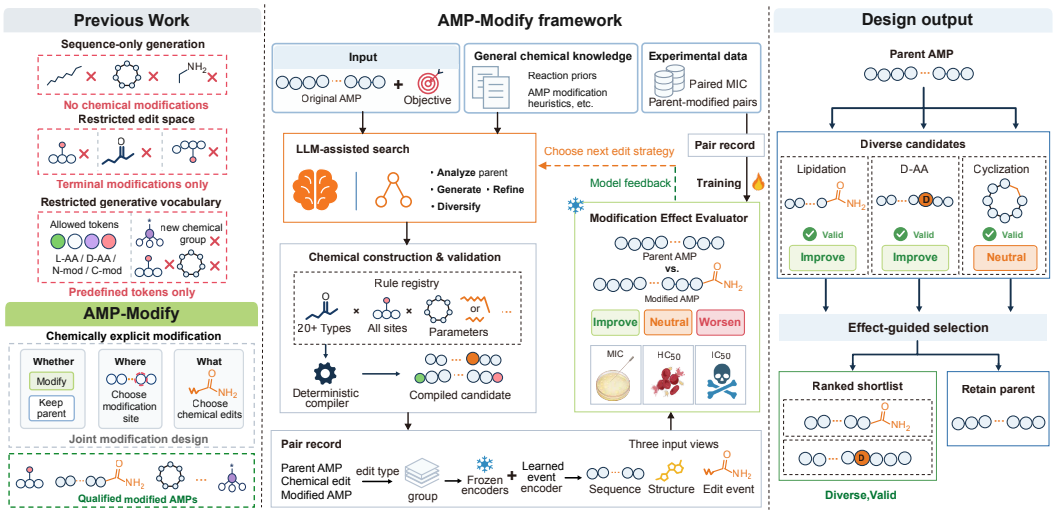

Figure 1: Overview of AMP-Modify. The framework jointly decides *whether*, *where*, and *what* to modify. An LLM-assisted search proposes edit strategies, a registry and compiler construct chemically valid candidates, and an evaluator scores each candidate relative to its parent. These scores guide the next search step. The output is a ranked shortlist, or the parent if no modification is predicted to help.

formulation offers limited support for designing chemical modifications of existing AMPs, including selecting modification sites and specifying the chemical structures to introduce.

Modification-aware AMP modeling. Several methods explicitly consider chemically modified peptides. AntiMPmod (Agrawal & Raghava, 2018) and HemoPImod (Kumar et al., 2020) predict antimicrobial or hemolytic properties of modified peptides, but treat the modified peptide as the prediction target rather than directly modeling how a specific edit changes its parent. Generative approaches have also expanded the accessible chemistry: Termini (Xu et al., 2026) explicitly designs N- and C-terminal modifications, while AMPGAN v3 (Jung et al., 2026) extends its generative vocabulary to D-amino acids and selected terminal modifications. These methods move AMP design beyond canonical sequences, but the supported modifications remain tied to particular modification families or vocabularies. They therefore do not jointly address site-specific chemical editing and parent-relative effect prediction across a broader modification space.

Relative-effect learning and guided chemical search. Relative molecular modeling has a long history in compound optimization, where matched molecular pairs relate structural transformations to changes in molecular properties (Turk et al., 2017). More recent pairwise learning methods such as DeepDelta (Fralish et al., 2023) directly learn property differences between molecular pairs rather than predicting each molecule independently. This perspective is particularly relevant to AMP modification, where the design question is whether a proposed chemical edit improves a specific parent peptide. However, existing molecular pairwise methods do not provide an AMP-specific representation and executable search space for site- and structure-specific peptide modifications. Tool-augmented LLM systems have separately shown that language models can coordinate chemical tools and use their outputs during iterative design (Bran et al., 2024; Boiko et al., 2023). AMP-Modify brings these ingredients together by representing chemical edits explicitly, learning their effects relative to the parent, and using this signal to guide exploration of an executable AMP modification space.

3 METHOD

We formulate AMP modification design as three coupled subproblems, defining the chemically valid search space, identifying promising modifications, and estimating their effects relative to the par-

ent. We define the search space through a set of chemical transformation rules and a molecular compiler, without requiring experimental training data (Section 3.2). Within this space, an LLM-assisted search uses chemical knowledge to choose the next region to examine (Section 3.4), while AMP-Modify-Eval learns from parent-modified pairs to predict the effects of candidate modifications (Section 3.3). The corresponding tools construct and evaluate candidates, and the accumulated evidence informs subsequent decisions (Section 3.4; Figure 1). The search returns a shortlist of supported candidates and may return none if no candidate meets the selection criteria.

3.1 Problem Formulation

A design task is $\tau = (p, f, o, R, K)$, where p is the parent peptide, $f \in \mathcal{F}$ is the target organism group, o specifies the desired activity direction, secondary objectives, and hard constraints, R is a versioned modification-rule registry, and K is the maximum shortlist size ($K = 20$ by default). We use seven organism groups: Enterobacteriaceae, Pseudomonadaceae, Staphylococcaceae, Enterococcaceae, other Gram-negative bacteria, other Gram-positive bacteria, and fungi. The experiments below are improvement-seeking; the pair model itself is trained on improve, neutral, and worsen.

The registry induces an operation space $\mathcal{A}_R(p)$. Each $a \in \mathcal{A}_R(p)$ is an explicit edit

$$a = (t, \ell, \phi), \tag{1}$$

with modification type t, site or site set ℓ, and structural parameters ϕ (for example a lipid chain, noncanonical residue, or cyclization pattern). A compiler maps a legal operation to a product m and retains it only if compilation succeeds. Each peptide state is represented as

$$x = (\mathbf{s}_x, \mathbf{c}_x, \mathbf{e}_x) \in \mathcal{X}, \tag{2}$$

where $\mathbf{s}_x$ is the residue sequence, $\mathbf{c}_x$ is the full-molecule isomeric SMILES, and $\mathbf{e}_x$ records the modification event. Sequence, structure, and event are complementary: many edits of interest leave the residue string unchanged, while the compiled graph and event identify attachment, topology, and stereochemistry.

The framework returns at most K compiled candidates that satisfy the hard chemical constraints and have an evaluated pair direction consistent with o. If none meet this criterion within the search budget, the shortlist is empty. That outcome indicates insufficient evidence for changing the parent.

3.2 Chemical Construction

The registry defines the executable space independently of the LLM. Each of the 20 construction types supplies applicability rules, allowed sites, a parameter domain, incompatibilities, a compiler template, and an event encoding. The types span five categories: terminal edits, residue and chirality substitutions, pendant groups, topology changes, and backbone or scaffold changes (Appendix B). Search therefore acts on normalized operations, not free-form molecules.

The LLM proposes a bounded edit strategy but does not emit SMILES. A deterministic generator intersects the strategy with $\mathcal{A}_R(p)$ and instantiates unseen operations. GENERATE starts from the parent, REFINE adds one compatible operation to a screened candidate, and DIVERSIFY favors underrepresented types, sites, and combinations. Previously tested signatures are excluded. Allocation details are in Appendix B.

Every instantiated operation passes through the same compiler and constraint checker before scoring. The compiler applies versioned residue and fragment templates to an RDKit molecular graph and returns canonical isomeric SMILES and a structured event. Invalid attachment atoms, valence, stereochemistry, site conflicts, ring closures, and failed SMILES round-trips cause rejection. The compiled event is the same object later read by the evaluator, so scoring is conditioned on the chemistry that was actually executed. Adding a type requires the three interfaces above and does not require retraining the evaluator. Compilation checks and the handling of already modified parents are in Appendix B.

3.3 Evaluator

General chemical knowledge and construction rules do not say whether a concrete edit helps a particular parent against a particular organism group. AMP-Modify-Eval is trained for that signal

and is frozen during search. For matched MIC values we set

$$\delta(p, m; f) = \log_2 \mathrm{MIC}_{m,f} - \log_2 \mathrm{MIC}_{p,f}. \tag{3}$$

A change of at least two-fold is labeled improve ($\delta \leq -1$) or worsen ($\delta \geq 1$); intermediate changes are neutral. The three-way label supervises the pair head; δ is retained for ranking analyses.

The model returns

$$g_{\mathrm{pair}} : \mathcal{X} \times \mathcal{X} \times \mathcal{F} \to \Delta(\{\mathrm{improve}, \mathrm{neutral}, \mathrm{worsen}\}), \tag{4}$$

$$g_{\mathrm{mic}} : \mathcal{X} \times \mathcal{F} \to \Delta(\{\mathrm{active}, \mathrm{inactive}\}), \tag{5}$$

$$g_{\mathrm{hc}}, g_{\mathrm{ic}} : \mathcal{X} \to \Delta(\{\mathrm{toxic}, \mathrm{safe}\}). \tag{6}$$

The pair head is the parent-relative effect; the remaining heads give absolute activity and toxicity context. For an improvement objective the ranking score is

$$s(p, m; f) = g_{\mathrm{pair}}(\mathrm{improve} \mid p, m, f). \tag{7}$$

Each state is encoded from three views: frozen ESM-2 650M (Lin et al., 2023) on the sequence, frozen PeptideCLM-2 (Feller et al., 2026) on the SMILES, and a learned encoder on the event. The branches are projected and fused by a two-layer Transformer. Sequence supplies scaffold context; the molecular graph preserves chain length, attachment, and topology; the event states what changed relative to the parent. Pair prediction uses the explicit comparison

$$[\mathbf{h}_p; \mathbf{h}_m; \mathbf{h}_m - \mathbf{h}_p; \mathbf{u}_f], \tag{8}$$

where $\mathbf{u}_f$ is the organism-group indicator. The MIC and toxicity heads read a single state. Search candidates are scored only after successful compilation. If a retrospective record has no reconstructable SMILES, the molecular branch can be zeroed (Appendix E).

Training uses

$$\mathcal{L} = \mathcal{L}_{\mathrm{mic}} + 0.05\,\mathcal{L}_{\mathrm{pair}} + 0.1\,\mathcal{L}_{\mathrm{hc}} + 0.1\,\mathcal{L}_{\mathrm{ic}}, \tag{9}$$

with each term applied only where a label exists. The corpus contains $8{,}145$ curated parent-modified relations from DBAASP (Pirtskhalava et al., 2021) and checked literature records, together with single-peptide MIC, HC50, and IC50 measurements. Pair labels come from matched MIC contrasts as above. Quantitative ranking uses the 980 Tier-A pairs with explicit parent identity, explicit modification chemistry, and matched parent/variant MIC. We split by connected components of the parent-variant graph so that related entities stay in the same fold. Dataset construction, thresholds, architecture, and optimization are in Appendix C.

3.4 Search and Selection

Search allocates a finite proposal budget over $\mathcal{A}_R(p)$. At iteration t the LLM receives the task and the scored history $\mathcal{H}_t$ and returns a structured strategy, not a molecule. A controller restricts the request to a feasible subset given evidence, remaining budget, and stopping rules; schema checks reject invalid plans (Appendix A). The generator then instantiates a batch of unseen operations, the compiler accepts or rejects each one, and the frozen evaluator scores every compiled product. The observation is appended to

$$\mathcal{H}_t = \{(m_i, E(m_i), a_i)\}_{i=1}^{N_t}. \tag{10}$$

Later strategies therefore condition only on compiled and scored outcomes. Analysis retrieves matched-pair evidence when the parent is first characterized or when progress stalls. Evaluator scores guide the next proposal through this history. They do not sample molecules, and the LLM cannot bypass compilation.

A candidate is qualified when it passes chemical constraints and $\arg\max g_{\mathrm{pair}}$ equals the requested pair direction. No extra threshold is placed on s or on the absolute activity gap. The pair head is the parent-relative gate and the feedback signal. Qualified candidates are then ordered by the change in predicted MIC-active probability,

$$\Delta P_{\mathrm{mic}}(p, m; f) = g_{\mathrm{mic}}(\mathrm{active} \mid m, f) - g_{\mathrm{mic}}(\mathrm{active} \mid p, f). \tag{11}$$

Thus s answers whether the edit is predicted to move in the requested direction, while ΔP_{mic} ranks candidates that already pass that directional test. Pair confidence, toxicity warnings, edit distance, and provenance remain available; HC50 and IC50 are advisory rather than hard filters. The shortlist is taken over the full history and truncated to K.

Search terminates under budget, quality, and convergence criteria detailed in Appendix A.

Table 1: Evaluator comparison on 980 Tier-A pairs. (A) External comparison. (B) Key ablations. Dashes mark missing outputs. Pair Spearman is not reported for the ablations. Five-seed variability is in Appendix D.

(A) External comparison

Method	Fam. bal.	Top-20%	Pair Sp	MIC F1	HC F1	IC F1
AMP-Modify-Eval	**0.378**	**0.699**	**0.403**	**0.732**	**0.799**	0.905
SeqOnly	0.068	0.306	0.093	0.711	0.755	**0.908**
Termini	0.211	0.561	0.302	0.652	0.730	–
AMPGAN	0.281	0.474	0.141	0.698	–	–
AMPGAN-e	0.217	0.566	0.266	0.708	–	–
HemoPImod	–	–	–	–	0.760	–

(B) Key ablations

w/o pair loss	0.053	0.383	–	0.748	0.806	0.901
w/o event	0.251	0.464	–	0.741	0.795	0.906
w/o SMILES	0.383	0.571	–	0.715	0.782	0.911

4 Experiments

4.1 Experimental Setup

Data. We train AMP-Modify-Eval on 8,145 curated parent-modified relations. These relations define the parent–variant graph for connected-component splits. The pair loss is applied to the 980 Tier-A pairs, for which the parent identity and modification chemistry are explicit and matched MIC measurements are available for both the parent and variant. Because the target of the pair head is the effect of a modification relative to its parent, these pairs provide the high-confidence subset on which the observed change in $\log_2 \mathrm{MIC}$ can be defined directly. We split the parent–variant graph by connected components so that chemically related parents and variants do not cross training and evaluation folds. Labeling details are given in Appendix C.

Metrics. We evaluate two complementary capabilities. For parent-relative effect prediction, family-balanced Spearman is the primary metric: it averages, across the seven organism groups, the correlation between the predicted improvement score $s(p, m; f)$ and the observed $-\Delta \log_2 \mathrm{MIC}$. We additionally report top-20% precision and Pair Spearman over all Tier-A pairs. MIC, HC50, and IC50 F1 evaluate the auxiliary absolute-property heads. These absolute heads provide activity and safety context, but they do not determine whether a particular modification improves its parent.

Baselines. SeqOnly retains only the ESM-2 sequence representation. Termini (Xu et al., 2026) scores compiled molecular structures using its SMILES-based predictor, whereas AMPGAN (Jung et al., 2026) represents modified peptides using the published residue-and-cap vocabulary; AMPGAN-e expands this vocabulary to 410 tokens while retaining the same token-based architecture. HemoPImod (Kumar et al., 2020) provides a modification-aware hemolysis baseline. All trainable baselines are evaluated on the same connected-component splits. Capability-matched evaluations are reported in Appendix D.

4.2 Parent-Relative Modification Effect Prediction

The evaluator is used during search to answer a specific question: given a parent peptide and a concrete chemical edit, is the modified peptide predicted to move in the requested direction? We therefore first test this parent-relative signal independently of the search procedure. The benchmark ranks held-out parent–modified pairs by the pair-head improvement score s; this evaluation should not be confused with final shortlist ordering, which uses ΔP_{mic} after the pair-direction gate.

AMP-Modify-Eval remains competitive on absolute AMP properties while improving parent-relative ranking (Table 1A). MIC F1 is 0.732, compared with 0.711 for SeqOnly, 0.698 for AMPGAN, and 0.652 for Termini; HC50 F1 is 0.799 versus 0.760 for HemoPImod, and IC50 F1 is 0.905.

Table 2: Literature modifications excluded from training. Pair is the five-seed majority vote vs. the published MIC direction. MIC cells are mean $P(\text{active})$ (parent → variant); green / red mark sign agreement, with $|\Delta P| < 0.01$ as neutral. Gray (C1) is unmarked. Dashes mark unevaluable cases. Sources and safety outputs are in Appendix E.

				MIC $P(\text{active})$			
ID	Edit	Lit. MIC	Pair	Ours	Ter.	AMPGAN	AMPGAN-e
C1	N-Ac	worsen†	improve†	0.15→0.16	0.10→0.05	0.17→0.20	0.17→0.18
C2	alt. D-AA	improve	improve ✓	0.36→0.38	0.55→0.55	0.53→0.92	0.43→0.76
C3	N-C8	improve	neutral ×	0.83→0.81	0.65→0.75	0.74→0.73	0.69→0.67
C4	staple	improve	improve ✓	0.71→0.72	–	–	0.65→0.67
C5	N-C8	improve	improve ✓	0.25→0.54	0.40→0.42	0.40→0.36	0.40→0.32
C6	L13→Ahx	neutral	neutral ✓	0.83→0.83	–	–	0.69→0.65
C7	N-PEG6	worsen	worsen ✓	0.82→0.61	0.71→0.72	–	–
C8	side-chain cycle	improve	improve ✓	0.82→0.76	0.71→0.37	–	–
C9	all-D	improve	neutral ×	0.73→0.90	0.63→0.63	0.80→0.72	0.81→0.76
C10	branched C8	improve	improve ✓	0.52→0.60	0.50→0.56	–	–

Existing modification-aware predictors introduce modification information with limited chemical resolution: Termini scores compiled SMILES with a terminal-oriented predictor, and AMPGAN uses a residue-and-cap token sequence. AMP-Modify-Eval reaches a family-balanced Spearman of 0.378, compared with 0.281 for AMPGAN, a top-20% precision of 0.699, compared with 0.561 for Termini, and a Pair Spearman of 0.403, compared with 0.302 for Termini. Expanding AMPGAN's discrete vocabulary does not close this gap (Appendix D). Absolute activity prediction alone is also insufficient: SeqOnly retains a MIC F1 of 0.711 but falls to 0.068 family-balanced Spearman, 0.306 top-20% precision, and 0.093 Pair Spearman.

The key ablations separate the relative objective from edit-specific inputs (Table 1B). Removing the pair objective drops family-balanced ranking from 0.378 to 0.053, even though MIC F1 rises to 0.748. A task-matched DeepDelta control learns a meaningful relative signal, but remains below AMP-Modify-Eval on the same 955 graph-available pairs (family-balanced Spearman 0.251 versus 0.372; Appendix D.3). Removing the event reduces family-balanced Spearman to 0.251 and top-20% precision to 0.464; removing SMILES leaves family-balanced Spearman nearly unchanged (0.383) but lowers top-20% precision to 0.571. Absolute activity prediction therefore does not recover the effect of a specific edit, and edit-specific representations improve the high-confidence ranking used to prioritize modifications. Remaining diagnostics are in Appendix D.

The Tier-A benchmark evaluates relative ranking on curated held-out pairs. We next ask whether the same signal is consistent with experimentally characterized modifications from studies excluded from evaluator training.

4.3 Validation on Experimentally Characterized Modifications

We next evaluate ten literature modifications whose corresponding modified entities and parent–modification pairs were excluded from training. This experiment complements the Tier-A benchmark in two ways: it tests chemically concrete modifications reported in independent studies, and it includes modification types whose structural detail is only partially representable by several existing predictors. C1 is retained for context but excluded from the MIC-direction summary because its source reports a CFU-based outcome rather than a matched MIC endpoint.

Across the nine MIC-labeled cases, the pair head agrees with the experimentally reported direction in seven. It recovers five of the seven reported improvements (C2, C4, C5, C8, and C10), while also identifying the neutral effect in C6 and the worsening effect in C7. These latter cases are useful controls because the literature set is enriched for modifications reported as beneficial; the evaluator does not simply label every published modification as an improvement. The two missed improvements, C3 and C9, are both predicted as neutral rather than worsen.

The auxiliary MIC head provides a complementary, but distinct, view. The direction of AMP-Modify-Eval's change in $P(\text{active})$ agrees with the literature direction in seven of the nine MIC-labeled cases. The corresponding counts are three of seven representable cases for Termini, one of four for AMPGAN, and two of six for AMPGAN-e. These counts are descriptive rather than a

Table 3: End-to-end search under a 100-slot budget per case. (A) Four settings over 700 slots. Unique-valid yield is over all slots; improve/valid is over unique-valid candidates; strict-improve yield is over the 700-slot budget. Dashes mark quantities that a setting does not define. (B) Nearest-candidate recovery as mechanism/site/parameter (✓ match, ∼ partial, – none, n/a inapplicable).

(A) Search settings

Method	Unique valid	Unique-valid yield	Improve / valid	Strict-improve yield
AMP-Modify	**628**	**89.7%**	**63.9%**	**57.3%**
w/o feedback	583	83.3%	58.1%	48.4%
Registry-Random	372	53.1%	46.5%	24.7%
Raw LLM	266	38.0%	–	–

(B) Recovery of withheld literature edits

Case	Withheld edit	AMP-Modify	Raw LLM
Canonical N-terminal lipidations			
C3	linear N-C8	∼ / ✓ / ✓	✓ / ✓ / ✓
C5	linear N-C8	✓ / ✓ / ✓	✓ / ✓ / ✓
C10	branched N-C8	✓ / ✓ / ✓	✓ / ✓ / ✓
Structurally specific edits			
C2	alternating D-AA	✓ / ∼ / n/a	– / – / n/a
C4	staple 33–37	✓ / – / ✓	✓ / – / ✓
C6	L13→Ahx	✓ / ✓ / ✓	– / – / –
C8	side-chain cycle	✓ / ✓ / ✓	– / – / –

direct comparison of parent-relative accuracy, because the baseline values are changes between two absolute activity predictions rather than explicit pair-effect outputs.

The literature cases also expose differences in representational coverage. AMP-Modify-Eval returns an output for all ten cases, subject to the reconstruction caveats in Appendix E, whereas Termini, AMPGAN, and AMPGAN-e are evaluable on eight, five, and seven cases, respectively. Enlarging the AMPGAN token vocabulary therefore increases coverage, but chemically specific transformations such as PEGylation, side-chain cyclization, and branched lipidation remain outside or only approximately represented by a residue-and-cap token sequence. Together, these results support the evaluator as a parent-relative signal for modification design.

4.4 Chemically Grounded Modification Search

End-to-end modification design poses a separate challenge from effect prediction. Before a candidate can be scored, the search must choose an edit, instantiate its site and structural parameters, and produce a chemically valid molecular structure. We evaluate search with only the parent peptide, target organism group, and improvement-seeking objective, withholding the literature modification until post hoc analysis. Seven constructive single-mechanism cases from Section 4.3 are used, with a fixed budget of 100 proposal slots per case. C1, C7, and C9 are excluded and C6 is retained as a reachability control; the inclusion rule is in Appendix F.3. Termini enumerates only terminal chemistry and AMPGAN's vocabulary does not cover this panel, so neither is used as a search baseline. The comparison is a Raw LLM that uses the same model without the registry or compiler.

Explicit construction substantially increases the fraction of the proposal budget that reaches evaluable chemistry (Table 3A). Across 700 slots, AMP-Modify returns 628 unique valid candidates (89.7%), with 72 unused slots and no invalid or duplicate proposals. The Raw LLM returns 266 unique valid candidates (38.0%) and spends 230 slots on invalid structures and 204 on duplicates. The main effect of the registry and compiler is thus not to make the language model itself chemically knowledgeable, but to ensure that proposed search effort is instantiated inside an executable chemical space.

Chemical validity alone, however, does not imply recovery of a relevant modification. The two methods behave similarly on familiar N-terminal lipidations: both exactly recover C5 and C10, and

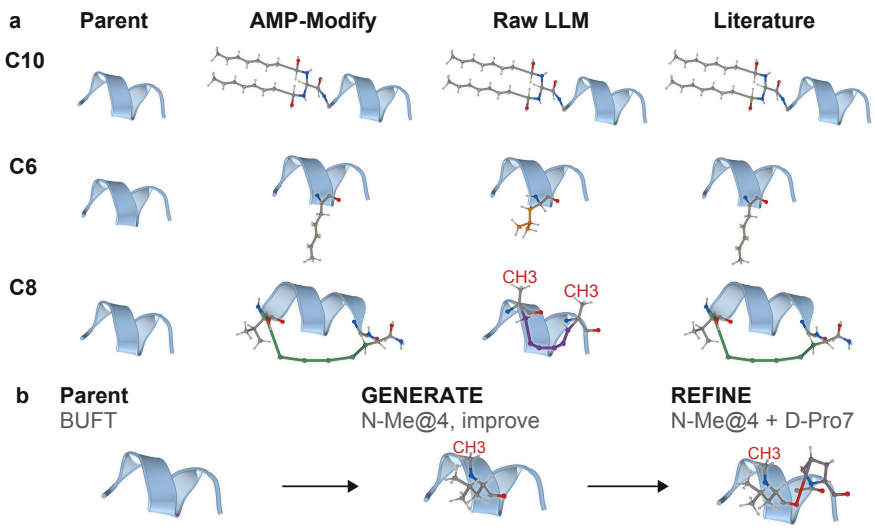

Figure 2: Representative outcomes from end-to-end chemical modification search. **a.** Exact recovery of C10, C6 (L13→Ahx), and C8 by AMP-Modify; the Raw LLM recovers only C10. **b.** One C10 feedback step: after N-Me@4 is predicted to improve, REFINE adds D-Pro7.

the Raw LLM exactly reproduces the linear N-C8 edit in C3 (Table 3B; Figure 2a). AMP-Modify instead reaches a branched C8 lipidation at the same N-terminal site, yielding a partial rather than exact match in C3. The distinction is larger for structurally specific edits. AMP-Modify exactly reaches the L13→Ahx substitution in C6 and the side-chain cyclization in C8, neither of which appears in the Raw-LLM candidate pool; C2 is partially recovered, while the precise stapling site in C4 remains difficult for both systems. These results separate two regimes: familiar modifications can often be proposed directly from language-model priors, whereas registered construction becomes useful when the target chemistry requires a specific site, topology, or structural parameter.

We also run the same seven cases under two additional settings, each with the same 100-slot budget (Table 3A). The w/o-feedback run keeps LLM-assisted exploration but does not write evaluator outputs back into the search state. Registry-Random samples the registered space without the LLM. Strict-improve yield is 57.3% for AMP-Modify, 48.4% without feedback, and 24.7% for Registry-Random; the improve/valid fractions are 63.9%, 58.1%, and 46.5%. These quantities count candidates that the frozen evaluator labels as improve; they are not measured MIC improvements. Each setting is one search per case. The Raw LLM is not assigned an improve yield, because it is scored only after the fact and is the construction control rather than a registered search setting.

Together, the ablations isolate the three components. Registered construction defines executable chemistry, LLM-assisted search improves budget use over uniform sampling, and evaluator write-back increases improve/valid and strict-improve yield. Registry-Random still constructs exact L13-Ahx and the side-chain cycle (Appendix F), so those recoveries are a property of the registered space. Figure 2b illustrates one step of the full system: after N-Me@4 is predicted to improve, REFINE adds D-Pro7 rather than reproducing the withheld literature modification.

5 Conclusion

We presented AMP-Modify, a framework for chemically explicit modification of existing antimicrobial peptides. Across held-out pairs, experimentally characterized modifications, and end-to-end search, it improves parent-relative effect ranking, reaches structurally specific edits that a Raw LLM misses, and reports a higher improvement yield when evaluator scores are written back into search. Therefore, the executable construction and parent-relative supervision support decisions over whether, where, and what to modify even when paired data are limited, without learning a single end-to-end generator.

6 Ethics Statement

This work is computational. No experiments involving pathogens, human subjects, or animals were performed, and no modification proposed here has been synthesized or assayed.

7 Reproducibility Statement

Code, the versioned modification registry, and the curated parent–modified relations are available at `https://anonymous.4open.science/r/AMP-Modify-BB63`. Appendix A specifies the controller state, decision schema, tool workflows, qualification rule, and default budgets. Appendix B lists the 20-type registry, the allocation policy, and compiler checks. Appendix C documents labeling, connected-component splits, architecture, and training. Appendix D reports capability-matched subsets, remaining ablations, and five-seed variability. Appendix E records literature endpoints and representation caveats. Appendix F gives the search protocol, case inclusion, and per-case pools.

AI Use Statement

In this work, we used generative AI tools for editing the paper to improve readability. We have reviewed all AI-assisted work. We have not used generative AI tools for generating synthetic datasets, developing theoretical or conceptual models, formulating mathematical claims, assisting with proofs, proposing or refining hypotheses, designing or providing feedback on methodology or experiments, implementing methods, translation, dataset cleaning or reformatting, qualitative or thematic data analysis, or interpreting results; the rest of the required research tasks are not applicable to this work. The authors independently verified all numerical results, scientific claims, methodological descriptions, and final interpretations against the underlying experiments and literature. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

References

Piyush Agrawal and Gajendra P. S. Raghava. Prediction of antimicrobial potential of a chemically modified peptide from its tertiary structure. *Frontiers in Microbiology*, 9:2551, 2018. doi: 10.3389/fmicb.2018.02551.

Francesca Albini, Barbara Biondi, Adriana Di Stasi, Andrea Schivo, Mario Mardirossian, Marco Scocchi, and Cristina Peggion. Fragments of cathelicidins PMAP-36 and BMAP-27 and their D-enantiomers: Effects of all D substitutions on structure, protease resistance and antimicrobial properties. *Bioorganic Chemistry*, 163:108715, 2025. doi: 10.1016/j.bioorg.2025.108715.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024. doi: 10.1038/s42256-024-00832-8.

Maria D'Accolti, Denise Bellotti, Emilia Dzień, Carlotta Leonetti, Silvia Leveraro, Valentina Albanese, Erika Marzola, Remo Guerrini, Elisabetta Caselli, Magdalena Rowińska-Żyrek, and Maurizio Remelli. Impact of C- and N-terminal protection on the stability, metal chelation and antimicrobial properties of calcitermin. *Scientific Reports*, 13(1):18228, 2023. doi: 10.1038/s41598-023-45437-0.

Hashem Etayash, Daniel Pletzer, Prashant Kumar, Suzana K. Straus, and Robert E. W. Hancock. Cyclic derivative of host-defense peptide IDR-1018 improves proteolytic stability, suppresses inflammation, and enhances in vivo activity. *Journal of Medicinal Chemistry*, 63(17):9228–9236, 2020. doi: 10.1021/acs.jmedchem.0c00303.

Hashem Etayash, Fione Yip, and Robert E. W. Hancock. Impacts of PEGylation and glycosylation on the biological properties of host defense peptide IDR1018. *Pharmaceutics*, 15(5):1391, 2023. doi: 10.3390/pharmaceutics15051391.

Aaron L. Feller, Maxim Secor, Sebastian Swanson, Claus O. Wilke, and Kristine Deibler. Scaling SMILES-based chemical language models for therapeutic peptide engineering. *Journal of Chemical Information and Modeling*, 66(14):8110–8122, jul 2026. doi: 10.1021/acs.jcim.6c00652. URL https://doi.org/10.1021/acs.jcim.6c00652.

Zachary Fralish, Ashley Chen, Paul Skaluba, and Daniel Reker. DeepDelta: predicting ADMET improvements of molecular derivatives with deep learning. *Journal of Cheminformatics*, 15(1): 101, 2023. doi: 10.1186/s13321-023-00769-x.

Robert E. W. Hancock and Hans-Georg Sahl. Antimicrobial and host-defense peptides as new anti-infective therapeutic strategies. *Nature Biotechnology*, 24(12):1551–1557, 2006. doi: 10.1038/nbt1267.

Yuchen Huan, Qing Kong, Haijin Mou, and Huaxi Yi. Antimicrobial peptides: classification, design, application and research progress in multiple fields. *Frontiers in Microbiology*, 11:582779, 2020. doi: 10.3389/fmicb.2020.582779.

Sheng Huang, Guoqi Su, Shan Jiang, Li Chen, Jinxiu Huang, and Feiyun Yang. New N-terminal fatty-acid-modified melittin analogs with potent biological activity. *International Journal of Molecular Sciences*, 25(2):867, 2024. doi: 10.3390/ijms25020867.

Jay Jung, Xiaohan Zhang, Shenghan Song, Mahmoud Sayedahmed, Chijian Xiang, Yunong Xu, Ahmed AbdelKhalek, Severin T. Schneebeli, Matthew J. Wargo, Jianing Li, and Safwan Wshah. Agentic discovery of non-canonical antimicrobial peptides with AMPGAN v3, 2026. arXiv preprint arXiv:2606.17127.

Vinod Kumar, Rajesh Kumar, Piyush Agrawal, Sumeet Patiyal, and Gajendra P. S. Raghava. A method for predicting hemolytic potency of chemically modified peptides from its structure. *Frontiers in Pharmacology*, 11:54, 2020. doi: 10.3389/fphar.2020.00054.

Ruifang Li, Xuxia Wang, Kedong Yin, Qingpeng Xu, Shiming Ren, Xueqin Wang, Zichao Wang, and Yanjie Yi. Fatty acid modification of antimicrobial peptide CGA-N9 and the combats against *Candida albicans* infection. *Biochemical Pharmacology*, 211:115535, 2023. doi: 10.1016/j.bcp.2023.115535.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Margit Mahlapuu, Joakim Håkansson, Lovisa Ringstad, and Camilla Björn. Antimicrobial peptides: an emerging category of therapeutic agents. *Frontiers in Cellular and Infection Microbiology*, 6: 194, 2016. doi: 10.3389/fcimb.2016.00194.

Bruno Mendes, Valeria Castelletto, Ian W. Hamley, and Glyn Barrett. D-amino acid substitution and cyclization enhance the stability and antimicrobial activity of arginine-rich peptides. *Microbiology*, 172(2):001657, 2026. doi: 10.1099/mic.0.001657.

Malak Pirtskhalava, Anthony A. Amstrong, Maia Grigolava, Mindia Chubinidze, Evgenia Alimbarashvili, Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E. Hurt, and Michael Tartakovsky. DBAASP v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic Acids Research*, 49(D1):D288–D297, 2021. doi: 10.1093/nar/gkaa991.

Naveenkumar Radhakrishnan, Sukumar Dinesh Kumar, Song-Yub Shin, and Sungtae Yang. Enhancing selective antimicrobial and antibiofilm activities of melittin through 6-aminohexanoic acid substitution. *Biomolecules*, 14(6):699, 2024. doi: 10.3390/biom14060699.

Yejiao Shi, Huixuan Chen, Sumeng Chen, Xiangxiang Kong, Zhinan Liu, Shaojie Wu, Xiang Li, Kai Zheng, Yan Xu, and Honggang Hu. N-terminal lipidation of buforin II generates a potent and stable peptide antibiotic. *Journal of Medicinal Chemistry*, 68(16):17624–17637, 2025a. doi: 10.1021/acs.jmedchem.5c01258.

Yejiao Shi, Gan Luo, Borui Zhen, Zhinan Liu, Sumeng Chen, Zhe Wang, Wuyuan Lu, Honggang Hu, and Xiang Li. Systematic all-hydrocarbon stapling analysis for cecropin A generates a potent and stable antimicrobial peptide. *Journal of Medicinal Chemistry*, 68(6):6372–6385, 2025b. doi: 10.1021/acs.jmedchem.4c02852.

Paulina Szymczak, Marcin Możejko, Tomasz Grzegorzek, Radosław Jurczak, Marta Bauer, Damian Neubauer, Karol Sikora, Michał Michalski, Jacek Sroka, Piotr Setny, Wojciech Kamysz, and Ewa Szczurek. Discovering highly potent antimicrobial peptides with deep generative model HydrAMP. *Nature Communications*, 14(1):1453, 2023. doi: 10.1038/s41467-023-36994-z.

Samo Turk, Benjamin Merget, Friedrich Rippmann, and Simone Fulle. Coupling matched molecular pairs with machine learning for virtual compound optimization. *Journal of Chemical Information and Modeling*, 57(12):3079–3085, 2017. doi: 10.1021/acs.jcim.7b00298.

Colin M. Van Oort, Jonathon B. Ferrell, Jacob M. Remington, Safwan Wshah, and Jianing Li. AMPGAN v2: machine learning-guided design of antimicrobial peptides. *Journal of Chemical Information and Modeling*, 61(5):2198–2207, 2021. doi: 10.1021/acs.jcim.0c01441.

Jing Xu, Marcelo D. T. Torres, Chen Li, Jian Li, Fuyi Li, Jiangning Song, and Cesar de la Fuente-Nunez. Modification-aware AI enables terminal chemical modifications for peptide design and discovers potent antimicrobials. *bioRxiv*, 2026. doi: 10.64898/2026.04.09.717597. bioRxiv 2026.04.09.717597, preprint.

Gen Zhou, Sugitha Janarthanan, Lianghong Chen, and Pingzhao Hu. MAC-AMP: A closed-loop multi-agent collaboration system for multi-objective antimicrobial peptide design. In *International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=iW1zfncsbc`.

A Search Controller, Decision Schema, and Stopping

The main text defines the edit, the compiler, the evaluator, and the search loop. This appendix records the controller that bounds LLM-assisted search: the allowed request types, the qualification rule, and the default budgets.

A.1 State Records and Decision Schema

The controller state stores the task, accumulated evidence, current hypotheses, candidate records, remaining budgets, and audit history. The task specifies the parent, target organism group, objectives, hard constraints, and generation requirements. Evidence includes sequence annotations, matched-pair records, tool observations, and uncertainty. Candidate records retain normalized edits, constraint status, evaluator outputs, objective change, edit distance, and the event-backed molecular representation.

At each step the LLM returns one admissible request together with a diagnosis, hypothesis update, tool plan, and, when generation is requested, a bounded edit strategy. It does not emit SMILES or assert candidate validity. The schema rejects references to nonexistent candidates or hypotheses, unavailable tools, budgets above the controller allocation, and tool plans inconsistent with the selected request. A malformed decision is given at most one retry. Only the controller can terminate the run.

The executor implements the following workflows:

$$\begin{aligned} \mathrm{ANALYZE}: &\ \text{sequence analysis} \rightarrow \text{matched-pair retrieval}, \\ \mathrm{GENERATE}, \mathrm{REFINE}, \mathrm{DIVERSIFY}: &\ \text{construction} \rightarrow \text{constraint check} \rightarrow \text{evaluation}, \\ \mathrm{VERIFY}: &\ \text{evaluation only}, \qquad \mathrm{STOP}: \ \text{no tool call}. \end{aligned} \tag{12}$$

ANALYZE retrieves matched-pair evidence when the parent is first characterized, and is reused when later scores contradict the current hypothesis or the search stagnates. The implementation may log that reuse as REANALYZE, but the workflow is the same. VERIFY re-scores an already compiled candidate with the frozen evaluator; it does not construct a new molecule and is distinct from compiler validation. If any required tool fails, no partial state transition is committed. On success, observations are stored in the evidence and audit histories, the iteration and tool budgets are debited, and candidate statistics are updated.

A.2 Qualification and Default Stopping Configuration

For a design task with requested pair direction o, a compiled candidate is qualified only when it passes the chemical constraint check and $\arg\max g_{\mathrm{pair}}$ agrees with o. The pair-head score s is this parent-relative gate and the ranking score on the Tier-A benchmark. We use ΔP_{mic} only as a secondary prioritization heuristic: after the pair head determines whether an edit is directionally beneficial relative to its parent, ΔP_{mic} favors candidates with larger predicted absolute activity gains within the qualified set. HC50 and IC50 predictions are retained as soft warnings and do not independently exclude a candidate.

The default satisfactory state requires baseline features, at least one supporting evidence item, the task-specified minimum number of qualified candidates, mean operation-set Jaccard diversity of at least 0.30, and mean uncertainty at most 0.50. The controller allows 10 iterations and 50 tool calls. A single generation, refinement, or diversification request receives at most 20, 10, or 15 proposal slots, respectively, under a global budget of 100 slots. Search also terminates when these budgets are exhausted or no generation action is executable. Two rounds without score progress trigger another analysis before budget-based termination. The returned list is truncated to the requested K. If no candidate qualifies, $\mathcal{S}(\tau) = \varnothing$. This indicates insufficient evidence for changing the parent within the search budget.

B Chemical Operations and Compilation

B.1 Modification Registry

The versioned generation registry contains the following 20 construction types:

1. C-terminal amidation;
2. N-terminal acetylation;
3. D-amino-acid substitution;
4. lipidation;
5. disulfide-bond formation;
6. head-to-tail cyclization;
7. lactam-bridge formation;
8. N-methylation;
9. ornithine substitution;
10. D-ornithine substitution;
11. diaminobutyric-acid substitution;
12. D-diaminobutyric-acid substitution;
13. diaminopropionic-acid substitution;
14. norleucine substitution;
15. hydrocarbon stapling;
16. other noncanonical-residue substitution;
17. PEGylation;
18. glycosylation;
19. glutamate-side-chain-to-N-terminus cyclization; and
20. branched lipidation.

These 20 types span the five categories in Section 3.2: terminal edits, residue and chirality substitutions, pendant groups, topology changes, and backbone or scaffold changes. Each rule declares its applicable sites, parameter domain, incompatibilities, macro-category, construction template, and event encoding. A new construction type must provide all three interfaces: applicability rules for enumeration, a molecular compiler template, and a structured event representation for the evaluator. This generation registry is independent of the representations used by the external scoring baselines in Appendix D.

B.2 Dynamic Allocation

The allocator uses a global budget of 100 previously unseen operation signatures. In the coverage phase it distributes slots across at least six applicable operation types and four macro-categories when the task permits. It enters calibration after sufficient coverage and collects at least three evaluated candidates per type before relying strongly on empirical outcomes. The optimization phase assigns more candidates to types supported by retrieved evidence and accumulated evaluator results, while retaining separate coverage, exploration, and combination buckets. By default, no operation type receives more than half of the global slots unless the applicable space itself makes that unavoidable. Deterministic ordering balances operation complexity, sequence region, physicochemical background, and previously successful sites.

GENERATE instantiates unseen operations from the parent under the current allocation. REFINE preserves the source candidate's normalized operations and adds one compatible operation under the maximum edit count. DIVERSIFY instead prioritizes operation sets and sites that are dissimilar to the evaluated pool. Both exclude duplicate signatures. Site checks cover the N-terminal amine, C-terminal carboxyl, relevant side-chain functional groups, stereocenters, and positions already consumed by a ring or attachment.

B.3 Compiler Validation

Compilation begins from the parent sequence or an exact event-backed parent structure. Operations are normalized before versioned residue and fragment templates edit the molecular graph. The

product is sanitized after each step and converted to canonical isomeric SMILES. The validator then checks valence, stereochemistry, attachment atoms, expected ring closure, a single connected component, parse round-trip, and stable canonicalization. A structure hash is stored with its operation provenance.

For an already modified parent, automatic extension is allowed only when its exact SMILES and existing events provide a reconstructable chemical context. Otherwise the parent can be screened but is not expanded. Likewise, chemistry without a registry template can be evaluated only when an external candidate provides complete isomeric SMILES and structured events; it is never generated from an unconstrained textual description.

C Evaluator Data and Training

C.1 Labels and Dataset Construction

Single-peptide records supervise absolute activity and toxicity. Under the primary binary protocol, $\mathrm{MIC} \leq 32\,\mu\mathrm{g\,mL}^{-1}$, or $\mathrm{MIC} \leq 10\,\mu\mathrm{M}$ when molecular weight is unavailable, is active; HC50 or $\mathrm{IC50} \leq 100\,\mu\mathrm{M}$ is toxic.

For parent–modified supervision, MIC measurements matched within the same organism group define the observed effect. After converting both values to μM,

$$\Delta \log_2 \mathrm{MIC} = \log_2(\mathrm{MIC}_{\mathrm{mod}}/\mu\mathrm{M}) - \log_2(\mathrm{MIC}_{\mathrm{parent}}/\mu\mathrm{M}). \tag{13}$$

Negative values correspond to a potency improvement. A decrease of at least two-fold is $\Delta \leq -1$ and is labeled *improve*; an increase of at least two-fold is $\Delta \geq 1$ and is labeled *worsen*; the remainder is *neutral*. Quantitative ranking uses this signed $\Delta \log_2 \mathrm{MIC}$.

Supervision comes from a cleaned DBAASP dump and manually checked literature records: $25{,}192$ peptide entities, $60{,}551$ organism-group MIC labels, $16{,}540$ HC50 measurements, $5{,}569$ IC50 measurements, and $8{,}145$ curated parent–modified relations. These $8{,}145$ relations define the parent–variant graph used for connected-component cross-validation. Pair supervision is narrower: only quality-A, parent-to-modified edges with a matched organism-group MIC enter $\mathcal{L}_{\mathrm{pair}}$ ($n = 980$). These are the same Tier-A pairs used for ranking; each pair is trained on when its connected component is in the training fold and evaluated when that component is held out. The remaining relations are not pair-effect labels.

MIC values in molar units are converted directly to μM; mass-concentration values are converted when molecular weight is available. An interval is replaced by its arithmetic midpoint, a left-censored value $< u$ by $u/2$, and a right-censored value $> u$ by $2u$. Non-positive values, incomparable units, and records without a valid conversion are excluded from quantitative pair supervision. We transform retained values to $\log_2(\mathrm{MIC}/\mu\mathrm{M})$ and take the median for each peptide and organism group. The 980 Tier-A pairs are those that remain after these conversion rules and the quality-A, matched-MIC criteria above.

To prevent direct or transitive parent–modified relations from crossing folds, canonical chemical entities are nodes in an undirected relation graph and curated modification pairs are edges. Entire connected components are assigned to five folds; preprocessing statistics and learned representations are recomputed from the training portion of each fold.

C.2 Architecture and Optimization

Frozen ESM-2 650M mean-pools the sequence and an MLP maps the result through $1280 \to 512 \to 192$. Frozen PeptideCLM-2 encodes the molecular SMILES and a second MLP uses $1024 \to 512 \to 192$. An approximately 102-dimensional event vector containing site, topology, mass change, scope, missingness, and named counts is mapped through $102 \to 160 \to 96$. The three branches are projected to $d = 256$ and combined with a CLS token and modality embeddings in a two-layer Transformer with eight attention heads, feed-forward width 1024, and dropout 0.25. Only the projections, fusion module, and prediction heads are trained.

The MIC head contains one binary classifier per organism group; HC50 and IC50 use toxic/safe classifiers. The pair head receives

$$[\mathbf{h}_p; \mathbf{h}_m; \mathbf{h}_m - \mathbf{h}_p; \text{family one-hot}], \tag{14}$$

a 775-dimensional vector, and applies a one-layer MLP of width 128 before the three-way output. The training loss is

$$\mathcal{L} = \mathcal{L}_{\text{mic}} + 0.05\mathcal{L}_{\text{pair}} + 0.1\mathcal{L}_{\text{hc}} + 0.1\mathcal{L}_{\text{ic}}. \tag{15}$$

MIC, pair, and toxicity loaders use batch sizes 512, 64, and 256, with pair sampling balanced across organism groups. AdamW uses learning rate 10^{-3} and weight decay 10^{-4} for at most 40 epochs with patience 8. The default run tracks the pair task for early stopping and checkpoint selection. When the pair loss is removed, that rule is switched to MIC F1.

Hadamard and cosine pair interactions, reverse-pair augmentation, a Δ auxiliary loss, potential and antisymmetric pair heads, five-way MIC classification, and $\log_2$ toxicity regression are evaluated only as controls. At inference, the evaluator is frozen. Main-table point estimates use seed 20260729. Search averages the five connected-component folds of that same seed; it is not a five-seed ensemble.

D Additional Evaluator Results

D.1 Metrics

Family-balanced Spearman is the unweighted mean of per-group Spearman correlations between $s(p, m; f)$ and $-\Delta \log_2 \text{MIC}$. A positive coefficient therefore means that higher s ranks larger potency improvements. Top-20% precision is the fraction of true improve pairs among the 20% of Tier-A pairs with the largest s. Pair Spearman is the same correlation computed over all 980 pairs, without group averaging. MIC, HC50, and IC50 F1 use the binary thresholds of Appendix C.

D.2 External baselines and AMPGAN-e

Termini, AMPGAN, AMPGAN-e, and HemoPImod in the main text are the public GitHub implementations of those methods, retrained on the same connected-component folds and seed 20260729. They are not the representation ablations that remove the event or SMILES branch.

AMPGAN uses the published 48-token residue-and-cap vocabulary. Tokens outside that set are blanked or dropped. AMPGAN-e expands this dictionary to 410 tokens by adding (i) sanitized N- and C-terminal names observed in our entity table and (ii) any residue character that appears in those sequences, plus an unknown token. The discriminator architecture is unchanged; only the one-hot vocabulary grows, and the model is fold-retrained. The expansion still encodes a peptide as one token string rather than a molecular graph or structured edit event. Stapling, Ahx chemistry, PEGylation, side-chain cyclization, and the branched C8 name therefore remain unsupported or only approximately represented (an `X` character is not the staple or Ahx structure).

Termini scores compiled SMILES with a Morgan-count logistic model. The capability-matched Termini subset keeps pairs whose N-terminus is free or acetylated and whose C-terminus is free or amidated ($n = 796$). The AMPGAN-vocab subset keeps pairs in which both peptides fit the native 48-token vocabulary ($n = 923$). Neither subset is retrained.

Table 4 recomputes pair metrics on these subsets. Restricting evaluation to chemistry supported by each baseline does not remove the ranking gap.

D.3 Pairwise Δ-learning control

DeepDelta (Fralish et al., 2023) is a pairwise Δ-learning control, not an AMP activity, hemolysis, or toxicity baseline. The question is whether training directly on parent–modified property differences already accounts for the relative ranking of AMP-Modify-Eval. The original DeepDelta protocol forms all cross-pairs after the split; that construction does not match organism-group MIC edges. We therefore retrain the official Chemprop v1.5.2 two-molecule interface on the observed parent–modified edges, using the same connected-component folds and seed 20260729.

Table 4: Pair metrics on capability-matched subsets (primary binary protocol, seed 20260729). Termini, AMPGAN, and AMPGAN-e are the same fold-retrained implementations as in the main text.

Subset	Method	Fam. bal.	Top-20%	Pair Sp	n
AMPGAN vocab	AMP-Modify-Eval	0.376	0.751	0.426	923
	AMPGAN	0.289	0.476	0.146	923
	Termini	0.239	0.557	0.311	923
Termini paper-cap	AMP-Modify-Eval	0.368	0.762	0.433	796
	Termini	0.251	0.537	0.308	796
	AMPGAN	0.342	0.438	0.150	796

Chemprop requires a molecular graph for both peptides. Of the 980 Tier-A pairs, 955 have reconstructable parent and variant SMILES. The remaining 25 lack a complete molecular graph for at least one endpoint (for example Fmoc, PEG, and metallocene edits) and are not scored by filling missing predictions with zeros. AMP-Modify-Eval can still score those pairs with the molecular branch zeroed (Section 3.3). DeepDelta numbers are therefore reported on the 955-pair subset, and AMP-Modify-Eval is re-evaluated on the same pairs. We score both methods with the family-balanced Spearman, top-20% precision, and Pair Spearman implementations used for Table 1. DeepDelta predicts $\Delta \log_2 \mathrm{MIC}$; the two Spearman coefficients compare this prediction with the observed Δ, and top-20% precision ranks pairs by $-\hat{\Delta}$. AMP-Modify-Eval continues to rank by $P(\mathrm{improve})$.

The published DeepDelta command (five epochs, sum aggregation, unshared message-passing networks) does not transfer to peptide-sized graphs. The control we report uses a shared D-MPNN and mean aggregation, the Chemprop settings that recover a usable relative signal on this corpus. We do not treat the unadapted run as the comparison, and we do not report MIC, HC50, or IC50 F1 for DeepDelta.

Table 5: Task-matched DeepDelta control on observed parent–modified edges (Chemprop v1.5.2, two-molecule interface, seed 20260729). AMP-Modify-Eval on 980 pairs is the Table 1 headline; the 955-pair row is the matched comparison.

Method	Fam. bal.	Top-20%	Pair Sp
DeepDelta, peptide-adapted	0.251	0.618	0.351
AMP-Modify-Eval, same 955 pairs	0.372	0.717	0.408
AMP-Modify-Eval, official 980 pairs	0.378	0.699	0.403

Table 5 shows that generic pairwise Δ-learning is a real relative-effect signal, but it does not account for the ranking of AMP-Modify-Eval. The gap between 0.372 and 0.251 is not attributed to explicit edit representation alone: the two models also differ in sequence encoding, organism-group conditioning, architecture, and auxiliary objectives. Evidence that edit-specific inputs contribute remains the *w/o event* and *w/o SMILES* rows of Table 1B.

D.4 Full evaluator ablations

Table 6 reports the complete evaluator ablation sweep. Table 1B repeats the three conceptual rows (*w/o pair loss*, *w/o event*, and *w/o SMILES*); the remaining variants stay here as architecture diagnostics.

Removing the pair objective produces the largest loss in family-balanced ranking, from 0.378 to 0.053, while MIC F1 increases from 0.732 to 0.748. Removing the modification-event representation lowers top-20% precision to 0.464. Removing the molecular input gives a different pattern: family-balanced Spearman is essentially unchanged in this seed (0.383 versus 0.378), while top-20% precision falls to 0.571. The pair objective and edit-specific inputs therefore affect parent-relative ranking in ways that are not captured by MIC F1 alone.

The remaining rows probe fusion, auxiliary heads, sampling, and coarser edit representations. Because the full sweep is reported for one training seed, the smaller differences are treated as diagnostic rather than as a stable ranking of architectural variants.

Table 6: Full systematic ablation of AMP-Modify-Eval (primary binary protocol, seed 20260729).

Variant	Fam. bal.	Top-20%	MIC F1	HC F1	IC F1
AMP-Modify-Eval	0.378	0.699	0.732	0.799	0.905
w/o SMILES	0.383	0.571	0.715	0.782	0.911
w/o event	0.251	0.464	0.741	0.795	0.906
w/o sequence	0.310	0.495	0.718	0.779	0.896
SeqOnly (no SMILES/event)	0.068	0.306	0.711	0.755	0.908
MLP fusion	0.271	0.526	0.733	0.788	0.902
w/o HC50 head	0.335	0.566	0.734	–	0.903
w/o IC50 head	0.313	0.418	0.743	0.792	–
w/o HC+IC	0.335	0.505	0.721	–	–
w/o pair loss	0.053	0.383	0.748	0.806	0.901
w/o family-balanced sampling	0.307	0.526	0.731	0.800	0.907
event → termini (keep SMILES)	0.368	0.643	0.734	0.800	0.912
event → coarse (keep SMILES)	0.341	0.628	0.733	0.787	0.910

D.5 Seed variability

Table 7 reports five training seeds on the same folds. Family-balanced Spearman has lower across-seed variation than top-20% precision (0.032 versus 0.081 standard deviation). The five-seed means are 0.350, 0.653, and 0.393 for family-balanced Spearman, top-20% precision, and Pair Spearman, respectively. Headline comparisons in Table 1 use only the matched seed-20260729 point estimates. The five-seed mean and standard deviation are a stability report and are not treated as a method row.

Table 7: AMP-Modify-Eval across five seeds (primary binary protocol, same folds).

Seed	Fam. bal.	Top-20%	Pair Sp	MIC F1	HC F1	IC F1
20260729	0.378	0.699	0.403	0.732	0.799	0.905
20260730	0.375	0.582	0.393	0.711	0.778	0.908
20260731	0.356	0.571	0.371	0.739	0.786	0.912
20260732	0.298	0.648	0.369	0.723	0.783	0.904
20260733	0.342	0.765	0.431	0.722	0.795	0.908
mean	0.350	0.653	0.393	0.725	0.788	0.907
std	0.032	0.081	0.026	0.011	0.009	0.003

E Experimental Literature Validation Details

E.1 Exclusion audit

The ten literature modifications were required to be absent from evaluator training. For each case we recorded the source DOI and PMID, the parent peptide, and the exact modified entity, then checked three inventories of the connected-component training corpus: published identifiers against the curated external-source DOI list and modeling-table PMIDs; the modified peptide name and sequence against the entity table; and the parent–modification pair against the pair-split table. A parent natural peptide may already occur in DBAASP. The case is excluded from training only when the DOI/PMID, the modified chemical entity, and the exact parent–modification pair are all absent. All ten cases pass this audit; rejected candidates failed because the PMID, modified entity, or pair was already in the corpus.

Table 2 in the main text reports the five-seed majority pair vote for these ten cases. Ties break as improve $>$ neutral $>$ worsen. C1 has no matched MIC endpoint and is shown for context but excluded from the nine-case MIC-direction summary. For C2–C10, the reference direction follows the same improve/neutral/worsen convention used for the Tier-A benchmark. Sources: C1 (D'Accolti et al., 2023), C2 (Mendes et al., 2026), C3 (Huang et al., 2024), C4 (Shi et al., 2025b), C5 (Li et al., 2023), C6 (Radhakrishnan et al., 2024), C7 (Etayash et al., 2023), C8 (Etayash et al., 2020), C9 (Albini et al., 2025), C10 (Shi et al., 2025a). AMPGAN-e is AMPGAN with the expanded 410-token vocabulary.

Table 8 records the reported experimental endpoint and the mean pair-class probabilities across the five evaluator seeds. Representation caveats are listed below; safety outputs are reported separately because the literature endpoints do not define a common HC50/IC50 benchmark.

Table 8: Experimental endpoints and mean pair-class probabilities for the ten literature modifications. C1 is not included in MIC-direction accuracy because its direction is based on a CFU endpoint rather than matched MIC.

ID	Target group	Literature endpoint	P_{imp}	P_{neu}	P_{wor}
C1	fungi	Plasma $t_{1/2}$ 18 $\rightarrow$ 135 min; *C. albicans* CFU weaker; no MIC/HC50/IC50	0.65	0.03	0.32
C2	Enterobact.	*E. coli* O157 MIC 125 $\rightarrow$ 15.62 μM ($\sim$8$\times$); trypsin-stable; no HC50/IC50	0.50	0.39	0.11
C3	Staphylococc.	*S. aureus* MIC 8 $\rightarrow$ 0.5 μg mL^{-1} (16$\times$); hemolysis rises above 4 μg mL^{-1}; no HC50/IC50	0.42	0.45	0.13
C4	Enterobact.	C-terminal staple MIC typically 3.13/3.13/6.25 μM; clinical GM 7.18 μM; parent weaker; CEC-0 HC50 174.2 μM; no IC50	0.67	0.27	0.06
C5	fungi	*C. albicans* 246.4 $\rightarrow$ 7.8 μg mL^{-1}; HC5 1004 $\rightarrow$ 318 μg mL^{-1}; TI 4.1 $\rightarrow$ 40.8; no IC50	0.91	0.06	0.03
C6	Staphylococc.	GM MIC 10.5 $\rightarrow$ 12.5 μM (neutral); HC10 2 $\rightarrow$ 64 μM; no IC50	0.37	0.52	0.10
C7	Staphylococc.	MRSA MIC 16 $\rightarrow$ 32 μg mL^{-1}; $\sim$20% hemolysis at 128 μg mL^{-1} $\rightarrow$ $\sim$0; no HC50/IC50	0.31	0.28	0.40
C8	Staphylococc.	MRSA 32.5 $\rightarrow$ 15.1 μM ($\sim$2$\times$); low PBMC toxicity; no HC50/IC50	0.51	0.39	0.10
C9	Enterobact.	*S. aureus* $>$64 $\rightarrow$ 16 μM; *E. coli* 4 $\rightarrow$ 2 μM; no HC50/IC50	0.44	0.49	0.07
C10	Enterobact.	BUFT MIC 128 $\rightarrow$ 8 μg mL^{-1}; hemolysis 1.37% $\rightarrow$ 1.59% at 256 μg mL^{-1}; no HC50/IC50	0.80	0.18	0.02

Table 9: Safety-related outputs for the literature cases (primary binary protocol; parent $\rightarrow$ variant). HC/IC cells are $P(\mathrm{toxic})$. Dashes mark methods that cannot represent that structure. Literature endpoints are heterogeneous and are not converted into a common external HC50/IC50 accuracy metric.

ID	Reported safety endpoint	Ours HC	Ours IC	Ter. HC	HemoPImod
C1	none	0.20$\rightarrow$0.22	0.07$\rightarrow$0.10	0.07$\rightarrow$0.20	0.52$\rightarrow$0.53
C2	none	0.47$\rightarrow$0.30	0.41$\rightarrow$0.32	0.24$\rightarrow$0.24	0.53$\rightarrow$0.53
C3	hemolysis rises $>$4 μg mL^{-1}	0.90$\rightarrow$0.93	1.00$\rightarrow$1.00	0.49$\rightarrow$0.58	0.74$\rightarrow$0.70
C4	parent HC50 174.2 μM	0.25$\rightarrow$0.46	0.95$\rightarrow$0.91	–	–
C5	HC5 1004 $\rightarrow$ 318 μg mL^{-1}	0.16$\rightarrow$0.78	0.28$\rightarrow$0.93	0.10$\rightarrow$0.21	0.43$\rightarrow$0.50
C6	HC10 2 $\rightarrow$ 64 μM	0.90$\rightarrow$0.92	1.00$\rightarrow$0.99	–	–
C7	hemolysis $\sim$20%$\rightarrow$0 at 128 μg mL^{-1}	0.85$\rightarrow$0.84	0.97$\rightarrow$0.86	0.24$\rightarrow$0.58	0.54$\rightarrow$0.45
C8	low PBMC toxicity; no HC50	0.85$\rightarrow$0.88	0.97$\rightarrow$0.99	0.24$\rightarrow$0.20	0.54$\rightarrow$0.68
C9	none	0.17$\rightarrow$0.39	0.62$\rightarrow$0.79	0.14$\rightarrow$0.14	0.39$\rightarrow$0.39
C10	hemolysis 1.37% $\rightarrow$ 1.59%	0.17$\rightarrow$0.72	0.67$\rightarrow$0.98	0.17$\rightarrow$0.37	0.33$\rightarrow$0.47

The reported safety endpoints are not standardized across studies (HC5, HC10, percent hemolysis at a fixed concentration, PBMC toxicity, or qualitative statements), and none of the ten studies reports a common IC50 endpoint. Accordingly, these outputs are provided for completeness rather than treated as an external safety benchmark.

E.2 Representation caveats

The ten literature cases are not represented with equal structural fidelity. C4 (hydrocarbon staple) and C6 (Ahx) have no compiled SMILES in this retrospective evaluator analysis and are scored from the available sequence/event inputs with a zero molecular vector. C7 uses a PEG6-acid approximation, C8 a head-to-tail-cycle SMILES proxy, and C10 a 2-ethylhexanoyl proxy. Counts in Table 2

are therefore *evaluable coverage*: a method can return a score, not that the input is an exact molecular reconstruction. These caveats apply to retrospective evaluator scoring; the end-to-end recovery analysis in Appendix F uses compiled search candidates.

F End-to-End Search Details

F.1 Protocol and scoring

Each case uses a fixed budget of 100 proposal slots. The published modification is withheld from every search prompt and is used only after the run to evaluate the unique-valid pool. AMP-Modify uses the registry, compiler, constraint checker, evaluator, and the controller of Appendix A. The Raw LLM control uses the same base language model and the same structured operation schema, but without the registry, compiler, or evaluator; normalization, compilation, and duplicate checks are applied only after the fact.

The Raw LLM is asked for five independent batches of 20 candidates in a structured operation schema, not free-form SMILES. Each returned item consumes one slot and is not replaced. A proposal is *valid* when its operations normalize, use only allowed types, and compile. It is *invalid* when the operation list is empty or malformed, a type is disallowed, or compilation fails. Two valid proposals are *duplicates* when they share the same semantic operation fingerprint after normalization; the first occurrence is unique-valid and later copies remain in the duplicate count. Unused AMP-Modify slots are unfilled generation quota. Across the seven cases this accounting is 266 unique-valid, 230 invalid, 204 duplicate, and 0 unused raw slots, or 38.0% unique-valid yield.

Recovery is scored against the published *increment*: chemical features already present in the parent are not counted again. For each method, the nearest unique-valid candidate maximizes the graded score $0.5\,\mathrm{mechanism} + 0.3\,\mathrm{site} + 0.2\,\mathrm{parameter}$. Mechanism is 0.5 F1 on normalized operation names, 0.3 F1 on registry types, and 0.2 F1 on macro-categories. Site is the Jaccard overlap of attachment positions when both sides have sites, and otherwise 1 if they share an attachment scope. Parameter is the fraction of target structural keys recovered (`lipid_id`, `staple_id`, `residue_id`, and related identifiers). If only the macro-category matches, site and parameter scores are halved. When several candidates obtain the same graded score, the first unique-valid proposal in generation order is kept. The main text reports the three components as marks rather than the scalar: 1 is ✓, a value in $(0, 1)$ is $\sim$, and 0 is –. An attribute that the increment does not define, such as a lipid identifier for C2, is n/a.

AMP-Modify, the w/o-feedback run, and Registry-Random are three search settings on the same seven cases and the same 100-slot budget. Each setting is run once per case. The w/o-feedback run keeps LLM-assisted exploration but does not write evaluator outputs back into the search state; candidates are scored after generation. Registry-Random removes the LLM and samples from the registered operation space.

F.2 Language-model configuration

LLM-assisted search and Raw-LLM calls request `gpt-5.6-sol` through a structured JSON schema, with a maximum of 4096 output tokens. Returned snapshots include both `gpt-5.6-sol` and `gpt-5.6-sol-2026-07-09`. The request does not set temperature, top-p, or a decoding seed; each case is one search. Provider defaults therefore apply and are not recorded in the request.

F.3 Case inclusion

The search analysis uses seven constructive single-mechanism cases from Section 4.3. C1 is omitted because it lacks a matched MIC endpoint, and C7 because the reported N-PEG6 edit worsens MIC under an improve-seeking objective. C9 is omitted because the literature modification is a global all-D inversion rather than a single registry operation; representing it requires stacking several D-amino-acid operations. C6 is retained although its reported effect is neutral because recovery here measures whether the search reaches the reported chemistry, not whether that chemistry should be selected as an improvement.

F.4 PER-CASE NOTES

Canonical N-terminal lipidations (C3, C5, C10). For C3, AMP-Modify reaches the N-terminal site and C8 chain but not the linear N-C8 combination: its pool contains branched N-C8, linear N-C10/N-C12, and Lys7-C8 candidates. The Raw LLM produces the exact linear N-C8 edit. Both methods recover C5 exactly. Both also recover the exact C10 increment, including `BRANCHED_LIPIDATION`, the N-terminus, `LYS_BIS_ACYL`, and `OCTANOYL_C8`. Per-case pool sizes are reported in Table 11.

Structurally specific edits (C2, C4, C6, C8). For C2, AMP-Modify generates the two reported D-amino-acid site pairs $\{2, 4\}$ and $\{6, 8\}$ separately, but never combines both into one candidate. The Raw LLM generates D-amino-acid edits only together with other operations and does not produce either reported pair. For C4, both methods reach the hydrocarbon-staple class and the reported `S5_I_I4` identifier, but neither reaches sites 33–37. AMP-Modify constructs exact L13-Ahx in C6, whereas the Raw LLM never proposes Ahx. It also constructs the exact glutamate-side-chain-to-N-terminus cycle in C8; the nearest raw candidate is a hydrocarbon staple. Per-case pool sizes are reported in Table 11.

F.5 RECOVERY UNDER THE ADDITIONAL SETTINGS

Table 10 reports the same recovery analysis for the w/o-feedback and Registry-Random settings. Registry-Random still constructs exact L13-Ahx and the glutamate side-chain cycle, so these transformations are already reachable from the registered space. Exact recovery alone therefore does not separate the three settings; proposal yield provides the more informative comparison.

Table 10: Nearest-candidate recovery under the w/o-feedback and Registry-Random settings. Marks follow Table 3B.

Case	Withheld edit	w/o feedback	Registry-Random
Canonical N-terminal lipidations			
C3	linear N-C8	✓/ ✓/ ✓	✓/ ✓/ ✓
C5	linear N-C8	✓/ ✓/ ✓	✓/ ✓/ ✓
C10	branched N-C8	✓/ ✓/ ✓	∼/ ✓/ ✓
Structurally specific edits			
C2	alternating D-AA	✓/ ∼/ n/a	✓/ ∼/ n/a
C4	staple 33–37	✓/ –/ ✓	∼/ –/ ✓
C6	L13→Ahx	✓/ ✓/ ✓	✓/ ✓/ ✓
C8	side-chain cycle	✓/ ✓/ ✓	✓/ ✓/ ✓

F.6 SLOT ACCOUNTING

Table 11 records unique-valid yield under a 100-slot budget.

Table 11: Per-case unique-valid yield under a 100-slot budget. Empty AMP-Modify slots are unused generation quota.

Case	AMP-Modify	Unused	w/o feedback	Registry-Random	Raw LLM
C2	74	26	59	38	21
C3	98	2	96	95	43
C4	95	5	95	38	51
C5	82	18	60	35	32
C6	100	0	98	95	48
C8	89	11	81	35	28
C10	90	10	94	36	43
Total	628	72	583	372	266

F.7 NI AND STRICT-IMPROVE YIELDS

The main text reports unique-valid yield, improve/valid, and strict-improve yield (Table 3A). For completeness we also record a non-worsening (NI) count. NI counts unique-valid candidates with pair-head $\arg\max \in \{\text{improve}, \text{neutral}\}$, and strict improve counts $\arg\max = \text{improve}$. Yields use the fixed 700 proposal slots as denominator, so unused, invalid, and duplicate slots are retained. AMP-Modify produces 624 NI and 401 strict-improve candidates (89.1% and 57.3%); w/o feedback produces 579 and 339 (82.7% and 48.4%); Registry-Random produces 368 and 173 (52.6% and 24.7%). These are class-based yields, not thresholds on objective_delta. The main-text improve/valid fractions 63.9%, 58.1%, and 46.5% use the unique-valid count as the denominator rather than the 700-slot budget.

F.8 RANKING OF EXACT LITERATURE EDITS

Search ranking is independent of the five-seed literature pair vote. After the pool is complete, unique-valid candidates are ordered by objective_delta, the same ΔP_{mic} used to order the qualified shortlist. The reported top-10 is this entire evaluated pool; it is not the improve-qualified shortlist, and it is not a five-seed ensemble rank. Search scoring averages the five folds of seed 20260729.

C6 makes this distinction explicit. Its literature direction and pair-head vote are both *neutral* ($P_{\text{neu}} = 0.52$), so the exact L13-Ahx edit is not improve-qualified. It is nevertheless rank 1 in the AMP-Modify unique-valid pool by objective_delta ($\Delta = +0.015$). Under both ablations the same exact edit is rank 2. Table 12 reports the AMP-Modify-pool rank of each exact recovered increment.

Table 12: Rank of the exact withheld increment in the AMP-Modify unique-valid pool, ordered by objective_delta. Pair arg max is the search evaluator label of that exact candidate.

Case	Exact in pool	Rank	Top-10	Pair arg max
C3	no	–	–	–
C5	yes	55	no	improve
C6	yes	1	yes	neutral
C8	yes	55	no	improve
C10	yes	84	no	improve

Under both additional settings, exact C8 is generated but remains outside the top 10 (ranks 31 and 23), and exact C10 under w/o feedback is rank 75. The experiments therefore distinguish two outcomes: reaching the reported chemistry and ranking it highly under the improve-seeking objective.