LEARNING FROM EXPERIENCE FOR MULTI-OBJECTIVE ANTIMICROBIAL PEPTIDE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Antimicrobial peptides (AMPs) are promising candidates for drug-resistant infections, but practical design must balance antimicrobial activity, pathogen-specific potency, hemolytic safety, and toxicity. Conflicts among these objectives make it difficult to identify peptides with balanced property profiles. Navigating such trade-offs motivates the reuse of prior design experience under different objective bottlenecks. However, existing methods do not explicitly leverage successful and failed design cases as reusable experience during iterative optimization. To address this gap, we introduce **PepMind**, an experience-guided framework for iterative multi-objective AMP optimization. PepMind converts evaluated designs into structured records containing sequences, objective outcomes, design rationales, and failure diagnoses. A learned router is used to select the category and amount of experience according to the current optimization state for targeted sequence refinement, with new outcomes subsequently added to memory. Experimental results consistently demonstrate PepMind's effectiveness: it achieves the best performance in 19 of 20 evaluation settings and improves four-objective hypervolume (HV) by an average of 67.1% over the strongest external baseline. Further analyses confirm cross-pathogen experience transfer and favorable predicted properties of the generated peptides. These results support persistent, state-conditioned experience reuse as an effective strategy for adaptive multi-objective peptide design. Our code is available at https://anonymous.4open.science/r/PepMind-D16C/.

1 INTRODUCTION

Antimicrobial resistance poses a growing global health threat (Murray et al., 2022; Naghavi et al., 2024). Antimicrobial peptides (AMPs) are promising alternatives because of their diverse mechanisms and activity against drug-resistant pathogens (Bucataru & Ciobanasu, 2024; Min et al., 2024). However, useful AMPs are difficult to identify within the vast peptide sequence space.

Artificial intelligence has advanced AMP design through data-driven sequence generation and property-guided optimization (Wan et al., 2024; Van Oort et al., 2021). However, practical AMP design requires balancing antimicrobial activity, target-specific potency, hemolytic safety, and toxicity safety (Yao et al., 2025; Wang et al., 2026). Figure 1(a) presents objective correlations derived from real AMP sequences collected from DBAASP database, a curated AMP database (Pirtskhalava et al., 2021). Activity and potency are positively related, but both tend to conflict with safety objectives (Abels et al., 2019; Hollmann et al., 2016). Because of these conflicts, improving one objective frequently degrades another, and no fixed modification rule is effective for every candidate. Therefore, a useful design strategy must explore the sequence space iteratively, adapting each round's modifications to the specific objectives that the current candidate still fails to satisfy.

Under such iterative optimization, outcomes from earlier rounds can provide useful experience for subsequent decisions: successful trials reveal strategies that resolve particular objective conflicts, whereas failed trials provide signals about ineffective or undesirable modifications (Shinn et al., 2023; Zhao et al., 2024; Song et al., 2024). The usefulness of such experience is inherently context dependent, as AMP properties are jointly influenced by sequence composition and physicochemical characteristics, and modifications that improve one objective may adversely affect another (Hollmann et al., 2016; Liu et al., 2023). Consequently, the same modification may be beneficial in one sequence context but ineffective or detrimental in another. This motivates maintaining evaluated

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.

outcomes as persistent experience and retrieving those most relevant to the current optimization state (Shinn et al., 2023; Zhao et al., 2024; Song et al., 2024). For example, an *E. coli* candidate may already satisfy activity and safety constraints but still fail on target-specific potency, in which case prior potency-related experience can provide more targeted guidance for refinement. Existing methods rarely organize evaluated outcomes as reusable experience for subsequent refinement (Murakami et al., 2023; Wang et al., 2025d;c; Zhou et al., 2026).

To address this limitation, we introduce PepMind, an experience-guided framework for AMP design. As illustrated in Figure 1(b), unlike previous methods that primarily use evaluation to guide the next search step, PepMind first diagnoses the current bottleneck and then uses a learnable router to adaptively retrieve relevant successful and failed experience for targeted sequence refinement. Newly observed outcomes are subsequently stored as structured experience for later rounds. In Figure 1(c), we compare PepMind with 11 baselines across four pathogen-specific tasks. PepMind achieves a mean four-objective HV of 0.292, representing a 65.0% improvement over the strongest baseline. Further analyses show steady gains across optimization rounds, effective experience transfer across pathogens, and biologically plausible candidate peptides.

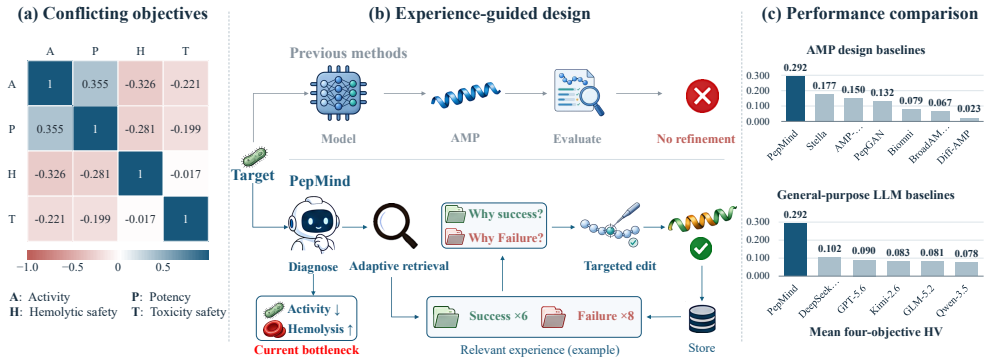

Figure 1: Motivation, design, and effectiveness of PepMind. (a) Objective correlations from DBAASP database reveal conflicts. (b) PepMind diagnoses the current bottleneck, adaptively retrieves relevant successful and failed experience for targeted edits, and stores new outcomes for later rounds. (c) Mean four-objective HV across four pathogens.

Our contributions are threefold.

1. We introduce PepMind, an experience-guided framework that converts successful and failed AMP design trials into persistent structured experience, preserving objective outcomes, design rationales, and failure diagnoses for reuse across optimization rounds.
2. We develop a state-conditioned dual-head router trained from counterfactual retrieval outcomes to adaptively select the category and amount of experience according to the current optimization state, enabling bottleneck-aware and targeted sequence refinement.
3. Across seven pathogen-specific tasks, PepMind achieves the best hypervolume in 19 of 20 main settings and improves four-objective hypervolume by 39.6%–92.3% over the strongest external baseline. Ablation and cross-pathogen transfer experiments further validate the effectiveness and reusability of the accumulated design experience.

2 Related Work

Deep Generative Models for AMP Design. Deep generative models provide an efficient way to explore the large peptide sequence space (Wan et al., 2024). Early methods learned sequence patterns from known AMPs, while later approaches incorporated property conditions to guide generation. HydrAMP, AMP-Designer, and AMPGen exemplify this progression through conditional models, transformer-based generation, and diffusion models (Szymczak et al., 2023; Wang et al., 2025b; Jin et al., 2025b). These approaches improve generation diversity and controllability over desired AMP properties. However, property-guided generation alone does not explicitly coordinate trade-offs among competing efficacy and safety objectives, motivating multi-objective optimization.

Multi-Objective Optimization for AMP Design. Recent AMP design methods increasingly consider multiple efficacy and safety requirements simultaneously (Liu et al., 2023; Xiong et al., 2026). Methods such as MODAN, HMAMP, MoFormer and MP2D address this problem through Bayesian optimization, hypervolume-driven learning, Pareto-based generation, and diffusion-based search (Murakami et al., 2023; Wang et al., 2025c;d; Torres et al., 2026; Kong et al., 2026). However, these methods mainly use evaluated candidates as optimization signals for subsequent search, leaving the accumulated information in successful and failed designs underexploited. PepMind instead accumulates these outcomes in long-term memory, enabling relevant design experience to be selectively reused across objectives and pathogens.

Agentic and Memory-Augmented Sequence Design. Agentic systems extend property-guided generation into iterative workflows that integrate generation, prediction, mutation, and evaluation (Guo et al., 2024). MAC-AMP employs multiple agents for closed-loop AMP optimization, while PABLO applies agentic black-box optimization to biological design, including AMP tasks (Zhou et al., 2026; Maus et al., 2026). More broadly, Biomni orchestrates diverse tools for biomedical research, whereas STELLA improves its capabilities through evolving reasoning templates and toolsets (Huang et al., 2026; Jin et al., 2025a). Memory-augmented agents such as Reflexion and ExpeL further demonstrate the value of experience from previous trajectories (Shinn et al., 2023; Zhao et al., 2024; Song et al., 2024). In AMP agents such as MAC-AMP and PABLO, experience is mainly used as feedback to guide subsequent optimization. PepMind instead treats experience reuse as a state-conditioned routing decision. For example, when potency fails, PepMind retrieves prior experience associated with similar potency bottlenecks to guide targeted refinement.

3 PRELIMINARY

3.1 MULTI-OBJECTIVE PROPERTIES AND EVALUATION METRICS

For peptide x and target pathogen s, we define four objectives, all to be maximized:

$$\mathbf{f}_s(x) = [A(x),\, H(x),\, T(x),\, P_s(x)]\,, \quad \begin{aligned} A(x) &= \mathrm{amPEPpy}(x), \\ H(x) &= 1 - \mathrm{HemoPI2}(x), \\ T(x) &= 1 - \mathrm{ToxinPred3}(x), \\ P_s(x) &= \left(1 + \frac{\overline{\mathrm{MIC}}_s(x)}{\mathrm{MIC}_s^{\mathrm{thr}}}\right)^{-1}. \end{aligned} \tag{1}$$

Specifically, antimicrobial activity, hemolytic risk, toxicity risk, and target-specific MIC are predicted using amPEPpy, HemoPI2, ToxinPred3, and APEX (Lawrence et al., 2021; Rathore et al., 2024; 2025; Torres et al., 2025). Their details are provided in Appendix A. All objectives are direction-aligned and bounded within $[0, 1]$, where larger values indicate better properties.

For each objective subset $S \subseteq \{A, H, T, P\}$ with $2 \leq |S| \leq 4$, we project the evaluated objective vectors onto S and compute the hypervolume (HV) of the resulting non-dominated set. This gives six two-objective, four three-objective, and one four-objective combination. The reference point is the zero vector of dimension $|S|$. A larger HV indicates better objective values and broader coverage of the trade-off space. Full details of the HV calculation are provided in Appendix E.

3.2 PROBLEM FORMULATION

Let $\mathcal{X}$ denote the set of peptide sequences composed of the 20 standard amino acids, with lengths between $L_{\min}$ and $L_{\max}$. At length $L_{\max}$, the search space contains $20^{L_{\max}}$ sequences. For a target pathogen s, the evaluators map each candidate $x \in \mathcal{X}$ to the objective vector $\mathbf{f}_s(x)$ defined in Eq. (1). Since the sequence space is discrete and the evaluators provide no gradients for sequence changes, we formulate AMP design as multi-objective optimization under a finite evaluation budget.

Starting from an empty archive $\mathcal{D}_0 = \varnothing$, a search policy π generates a batch $\mathcal{Q}_r$ at round r:

$$\mathcal{Q}_r = \pi(s, \mathcal{D}_{r-1}), \qquad \mathcal{D}_r = \mathcal{D}_{r-1} \cup \{(x, \mathbf{f}_s(x)) \mid x \in \mathcal{Q}_r\}. \tag{2}$$

Here, $\mathcal{D}_r$ records what has been evaluated: sequences and their objective values. PepMind augments this archive with structured experience that preserves why prior designs succeeded or failed, and

learns what category and amount of experience to retrieve at each round. The search is subject to:

$$\sum_{r=1}^{R} |\mathcal{Q}_r| \leq B, \tag{3}$$

where R is the number of optimization rounds and B is the total evaluation budget. Both newly generated sequences and refined sequences evaluated during the search count toward this budget.

For a candidate with complete and valid scores, the violation vector identifies unmet objectives:

$$\boldsymbol{\delta}_s(x) = [\mathbb{I}(A(x) < \tau_A), \mathbb{I}(H(x) < \tau_H), \mathbb{I}(T(x) < \tau_T), \mathbb{I}(P_s(x) < \tau_P)]\,. \tag{4}$$

Here, $\mathbb{I}(\cdot)$ equals one when its condition holds and zero otherwise, and $\tau_A, \tau_H, \tau_T, \tau_P$ are the corresponding feasibility thresholds. Let $\mathcal{V}_s(x)$ denote the subset of $\{A, H, T, P\}$ marked as violated. A candidate is feasible when $\boldsymbol{\delta}_s(x) = [0, 0, 0, 0]$, meaning that all four objectives satisfy their corresponding thresholds. Details of thresholds are provided in Appendix B.

4 METHOD

4.1 OVERVIEW OF PEPMIND

PepMind addresses target-specific AMP optimization by reusing experience accumulated during search. Its core design combines structured long-term memory with state-conditioned adaptive routing and experience-guided refinement. As shown in Figure 2(a), PepMind jointly optimizes four primary objectives while also considering novelty and structural reliability. At each round, the current optimization state, including objective signals and search progress, is passed to a learned dual-head router that adaptively selects the memory category and retrieval amount (Figure 2(b)). Relevant records are then filtered, ranked, and retrieved through top-k selection to construct refinement guidance. The target pathogen and objective profile are combined with the retrieved experience to build this guidance, which conditions the refinement policy π to determine whether candidates should be adjusted, redesigned, or reused (Figure 2(c)). The resulting candidates are evaluated by multiple predictors, and promising AMP candidates are retained for subsequent optimization. New outcomes are then stored in long-term memory as successful or failure-associated experience for future rounds (Figure 2(d)). For example, when an *E. coli* candidate satisfies activity and safety but fails potency, PepMind retrieves relevant experience to guide potency improvement in the next refinement.

4.2 STRUCTURED LONG-TERM EXPERIENCE MEMORY

PepMind stores evaluated designs in a structured memory:

$$\mathcal{M} = \{m_i\}_{i=1}^{N_{\mathcal{M}}}, \tag{5}$$

where $N_{\mathcal{M}}$ is the number of stored records. We use $\mathcal{M}_{r-1}$ to denote the memory available before round r. Unlike $\mathcal{D}_{r-1}$, which stores sequences and objective vectors, $\mathcal{M}_{r-1}$ retains the context and reasoning. Each record contains six fields: $m_i = (\mathcal{I}_i, \mathcal{O}_i, \mathcal{E}_i, \mathcal{R}_i, n_i, \mathcal{U}_i)$.

Example (abbreviated)	
$\mathcal{I}_i$: *E. coli*, improve activity	$\mathcal{O}_i$: `KKL...LG`; K/L aids membrane binding
$\mathcal{E}_i$: $A = 0.64 < \tau_A, \ldots$	$\mathcal{R}_i$: increase cationicity or hydrophobic moment
n_i: activity failure	$\mathcal{U}_i$: label $\{A\}$; round 3; retrieved twice

The context field $\mathcal{I}_i$ specifies the target pathogen and design constraints. The design field $\mathcal{O}_i$ stores the generated sequence together with the rationale for its design. The evaluation field $\mathcal{E}_i$ stores the predictor outputs and threshold checks. The decision field $\mathcal{R}_i$ records the subsequent action and the reason for that decision. The summary n_i briefly describes the overall design outcome. The auxiliary field $\mathcal{U}_i$ stores failure labels, generation metadata, and experience reuse statistics.

Let x_i and s_i denote the sequence and source pathogen of record m_i. The indicator $v_i \in \{0, 1\}$ specifies whether all four predictor outputs are complete and valid. Following Eq. 4, a valid record with $\boldsymbol{\delta}_{s_i}(x_i) = [0, 0, 0, 0]$ provides successful experience. Otherwise, each entry equal to 1 indicates a violated objective in the order A, H, T, and P, and the corresponding failure categories

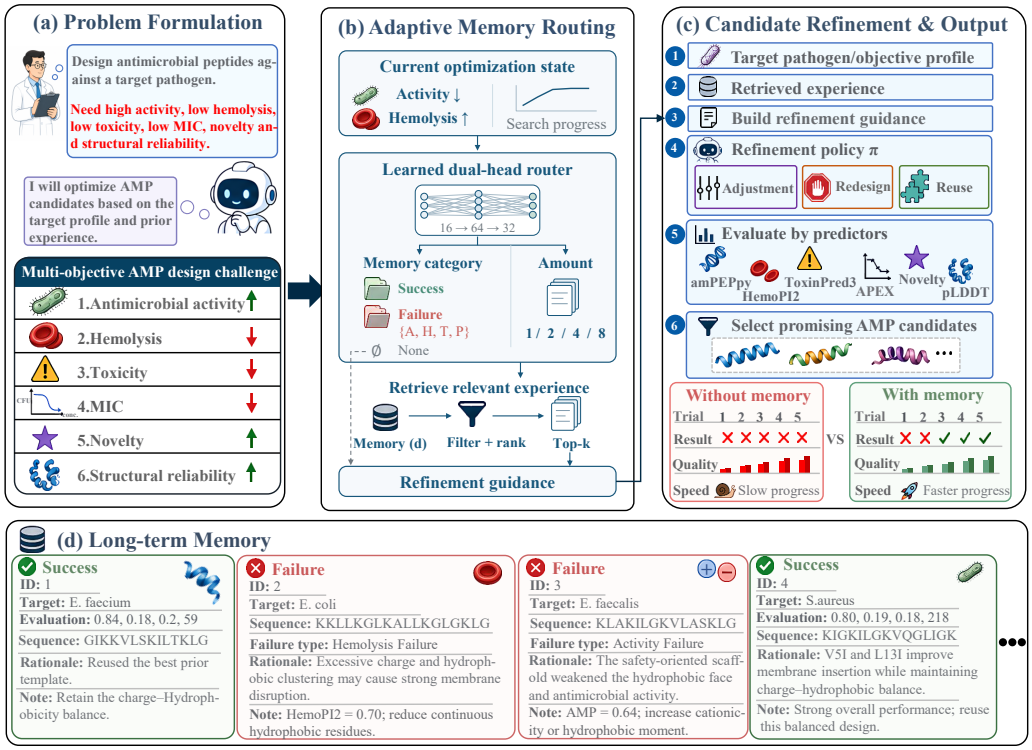

Figure 2: Overview of PepMind. (a) Target-specific AMP design under multiple property requirements. (b) State-conditioned adaptive memory routing and retrieval for constructing refinement guidance. (c) Policy-based candidate refinement, evaluation, and selection guided by retrieved experience. (d) Structured long-term memory of successful and failure-associated design experiences.

form $\mathcal{V}_{s_i}(x_i) \subseteq \{A, H, T, P\}$. A record may therefore belong to multiple failure categories. For example, $\boldsymbol{\delta}_{s_i}(x_i) = [1, 0, 0, 1]$ indicates failures in activity and target-specific potency, giving $\mathcal{V}_{s_i}(x_i) = \{A, P\}$. **Failure labels are essential for targeted refinement, as retrieving relevant failure categories helps avoid repeated errors and improves search efficiency.**

4.3 Adaptive Memory Routing and Retrieval

Before round r, a trained router selects experience from $\mathcal{M}_{r-1}$. Its training labels are constructed through counterfactual replay. For each optimization state, we replay routing actions with different memory categories and retrieval sizes, together with a no-memory action. Each action is evaluated by the four-objective HV improvement, with a penalty for retrieving more records. These scores are converted into probabilities that indicate which routing actions would have been more beneficial at that optimization state. The probabilities are then aggregated over retrieval sizes and memory categories to form soft labels for the type and size heads. Training details are provided in Appendix C. The input $\mathbf{h}_r \in \mathbb{R}^{16}$ summarizes objective scores, threshold violations, and search progress before round r. A shared backbone produces logits $\mathbf{z}_{\text{type}}$ and $\mathbf{z}_{\text{size}}$ for two categorical heads:

$$\mathbf{p}_{\text{type}} = \text{softmax}(\mathbf{z}_{\text{type}}), \qquad \mathbf{p}_{\text{size}} = \text{softmax}(\mathbf{z}_{\text{size}}). \tag{6}$$

The router makes two decisions: which category of experience to retrieve and how many records to use. The category can be successful designs, failures associated with activity, hemolysis, toxicity, or target-specific potency. When memory is used, the retrieval size is $k_r \in \{1, 2, 4, 8\}$, and only categories with available records are considered. Based on the selected category c_r, PepMind filters the available memory $\mathcal{M}_{r-1}$ to retain records with the corresponding success or failure label, forming $\mathcal{M}_{r-1}^{c_r}$. The retrieved set is:

$$\mathcal{K}_r(x) = \text{TopK}_{k_r}\left(\mathcal{M}_{r-1}^{c_r}; \rho(m_i, x)\right), \tag{7}$$

where $\rho(m_i, x)$ measures how relevant an eligible record is to the current candidate based on category relevance and physicochemical compatibility. Its complete definition is provided in Appendix D. PepMind ranks the eligible records by this score and retrieves the top k_r records, or all available records if fewer than k_r exist. No records are retrieved when the router selects no memory. In short, the router decides which category of experience to use and how much to retrieve, while the relevance score determines which specific records are most useful for the current candidate.

4.4 Experience-Guided Decision Making and Candidate Refinement

For an infeasible candidate x evaluated before round r, PepMind summarizes the records retrieved from $\mathcal{M}_{r-1}$, $\mathcal{K}_r(x)$ into:

$$\mathcal{G}_r(x) = \left(\mathcal{G}_r^{\text{feas}}(x), \mathcal{G}_r^{\text{fail}}(x)\right), \tag{8}$$

where $\mathcal{G}_r^{\text{feas}}(x)$ summarizes useful patterns from successful designs that may help improve the current candidate, while $\mathcal{G}_r^{\text{fail}}(x)$ summarizes relevant failure cases to avoid repeating similar mistakes.

The refinement decision is:

$$d_r(x) = \pi_{\text{dec}}\left(s, x, \mathbf{f}_s(x), \boldsymbol{\delta}_s(x), \mathcal{G}_r(x)\right). \tag{9}$$

Here, π_{dec} is the LLM-based refinement component of the search policy π defined in Section 3.2. Using the fixed prompt template in Appendix K, it specifies sequence changes and their rationale to improve violated objectives while preserving satisfied ones. Refined candidates enter $\mathcal{Q}_r$ and are evaluated. Feasible candidates undergo additional novelty and structural screening. The evaluation results are added to $\mathcal{D}_r$, while complete design experiences are stored in $\mathcal{M}_r$ and the usage statistics of retrieved records are updated to support subsequent retrieval.

5 Experiments

This section presents a series of experiments designed to address the following research questions:

RQ1: How effectively does PepMind jointly optimize multiple objectives in AMP design?

RQ2: How do long-term memory and adaptive routing contribute to the optimization process?

RQ3: Can optimization experience be transferred across pathogen-specific AMP design tasks?

RQ4: Do the generated peptides exhibit biological plausibility and structural reliability?

5.1 Implementation details

We compute HV for all 11 objective subsets and report five representative subsets: AH, AT, AHT, HTP, and $AHTP$, covering activity–safety, potency–safety, and full four-objective trade-offs. Comparisons include biomedical research agents, conventional AMP design methods (Wang et al., 2025b; Li et al., 2025; Wang et al., 2024; Tucs et al., 2020), general-purpose LLMs, and PepMind's sequential variant, which optimizes one objective per round. As a complementary metric, an external LLM scores each sequence from 1–10 based on activity, safety, structural plausibility, and developability using a fixed rubric. The full prompt is provided in Appendix K.1. All methods use the same budget of 300 candidates per task. PepMind's memory is initialized empty for each task and updated only with candidates evaluated within that run. HV is computed in the original objective space for the main comparison, while ablations use shared min–max bounds across all variants. The optimization agent uses DeepSeek-V4-Pro, requiring approximately 2,142 API calls across 998 optimization rounds at an estimated total cost of \$0.74–\$5.03.

5.2 Effectiveness of Joint Multi-Objective Optimization (RQ1)

Table 1 shows that PepMind achieves the highest HV in 19 of 20 settings. The sequential variant leads only on AT for *K. pneumoniae*. PepMind ranks first in AHT, HTP, and $AHTP$ in all four pathogens, demonstrating consistent advantages when efficacy and safety are evaluated jointly. Under $AHTP$, PepMind outperforms the strongest external baseline by 83.7%, 39.6%, 52.7%, and 92.3% on *E. coli*, *P. aeruginosa*, *K. pneumoniae* and *S. aureus*. To assess potential overfitting of

Table 1: Hypervolume and external LLM evaluation scores on four AMP design tasks. The best and second-best results are shown in bold and underlined, respectively.

	Hypervolume					LLM Evaluation			
Method	$\mathbf{HV}_{AH}$	$\mathbf{HV}_{AT}$	$\mathbf{HV}_{AHT}$	$\mathbf{HV}_{HTP}$	$\mathbf{HV}_{AHTP}$	**GPT-5.6**	**GLM-5.2**	**Qwen-3.5**	**Kimi-2.6**
E. coli									
PepMind	**0.6053**	**0.6355**	**0.4772**	**0.3860**	**0.3112**	**8.52**	**8.41**	7.45	**7.30**
PepMind (Seq.)	0.3786	0.5693	0.2476	0.2157	0.1878	8.36	8.01	6.66	7.20
Biomni	0.4894	0.2849	0.2300	0.1251	0.0758	7.41	7.60	6.99	6.67
STELLA	0.4956	0.4687	0.3552	0.2590	0.1694	7.63	7.24	6.61	6.39
AMP-Designer	0.2608	0.3910	0.1277	0.1505	0.1202	7.51	7.19	**7.58**	6.88
BroadAMP-GPT	0.1826	0.1884	0.0532	0.0743	0.0480	3.98	3.34	4.58	4.53
PepGAN	0.2855	0.3479	0.1298	0.1465	0.1121	6.62	6.61	6.37	6.30
Diff-AMP	0.2106	0.1431	0.0577	0.0317	0.0166	5.46	5.39	6.13	5.45
P. aeruginosa									
PepMind	**0.6430**	**0.6165**	**0.5037**	**0.3800**	**0.2990**	**8.32**	**8.05**	**7.83**	6.70
PepMind (Seq.)	0.4921	0.5843	0.3434	0.2676	0.2241	8.21	8.03	6.55	**6.94**
Biomni	0.5079	0.3011	0.2408	0.1208	0.0767	7.43	7.60	7.19	6.72
STELLA	0.4915	0.5007	0.3713	0.3232	0.2142	7.72	7.36	6.71	6.63
AMP-Designer	0.3695	0.4559	0.2246	0.2632	0.1975	7.18	6.78	7.40	6.76
BroadAMP-GPT	0.1901	0.3947	0.1114	0.1252	0.0843	3.98	3.37	4.38	4.66
PepGAN	0.2629	0.5092	0.1783	0.1842	0.1383	6.30	6.17	6.14	6.05
Diff-AMP	0.2300	0.1442	0.0679	0.0461	0.0225	4.55	4.72	5.86	5.09
K. pneumoniae									
PepMind	**0.6532**	0.5459	**0.4382**	**0.3018**	**0.2456**	**8.22**	7.59	**7.39**	**7.24**
PepMind (Seq.)	0.4971	**0.6115**	0.3706	0.2617	0.2146	8.11	**8.02**	7.00	7.19
Biomni	0.4908	0.2794	0.2282	0.1344	0.0808	7.44	7.65	7.17	6.71
STELLA	0.4937	0.4778	0.3628	0.2473	0.1608	7.68	7.28	6.67	6.55
AMP-Designer	0.2759	0.4392	0.1526	0.1853	0.1471	7.12	6.78	7.17	6.52
BroadAMP-GPT	0.1831	0.2677	0.0744	0.1016	0.0670	4.36	3.56	4.68	4.79
PepGAN	0.2870	0.4170	0.1603	0.1886	0.1409	6.52	6.34	6.01	6.07
Diff-AMP	0.2122	0.0952	0.0444	0.0467	0.0213	4.90	5.08	5.91	5.24
S. aureus									
PepMind	**0.6476**	**0.6872**	**0.5465**	**0.3830**	**0.3119**	**8.32**	**7.94**	**7.65**	**7.49**
PepMind (Seq.)	0.5976	0.6741	0.5240	0.2886	0.2218	8.30	7.80	7.14	7.46
Biomni	0.4930	0.2722	0.2219	0.1373	0.0830	7.66	7.68	7.04	6.66
STELLA	0.4948	0.4828	0.3661	0.2485	0.1622	7.36	6.91	6.36	5.91
AMP-Designer	0.2264	0.4799	0.1366	0.1680	0.1336	6.87	6.70	7.48	6.48
BroadAMP-GPT	0.1828	0.2619	0.0724	0.1039	0.0687	4.19	3.59	4.44	4.74
PepGAN	0.2346	0.4669	0.1412	0.1741	0.1350	6.69	6.64	6.47	6.29
Diff-AMP	0.2194	0.1463	0.0580	0.0587	0.0325	5.23	5.27	6.22	5.18

the predictors used during optimization, we performed external LLM and independent predictor evaluations. The LLM evaluation shows that **PepMind achieves the highest score in 13 of 16 combinations of pathogen and evaluator, with its sequential variant leading in two additional combinations.** The independent predictors evaluate the same selected candidates without reselection and show that PepMind largely retains its advantage, achieving the highest overall HV and leading performance in 9 of 10 objective combinations (Appendix G).

These results support joint optimization. The sequential variant's advantage on AT does not extend to AHT or $AHTP$, highlighting the importance of coordinating competing objectives during refinement. Rather than optimizing objectives in isolation, PepMind conditions refinement on the joint violation state and reuses experience matched to the current bottleneck, helping improve unmet objectives while preserving already satisfied ones. This mechanism is particularly useful for higher-order objective combinations, where efficacy, safety, and target-specific potency must be coordinated simultaneously. LLM baselines and complete objective subset results are provided in Appendices G.1 and F. Complete harmonic mean results are reported in Appendix H.

5.3 Contribution of Long-Term Memory and Adaptive Routing (RQ2)

Evolutionary Optimization Dynamics under Full Memory. Figures 3(a)–(b) illustrate PepMind's optimization over 30 rounds under full memory, divided into early (1–10), middle (11–20), and late (21–30) stages. From the early to late stage, stage-wise mean scores increase from 0.7442 to 0.7959 for activity, from 0.7686 to 0.7850 for non-toxicity, and from 0.7775 to 0.8106 for target-specific

potency. Hemolytic safety also improves modestly. In Fig. 3(b), candidates shift toward regions with higher potency and safety. These results suggest balanced improvement across multiple objectives.

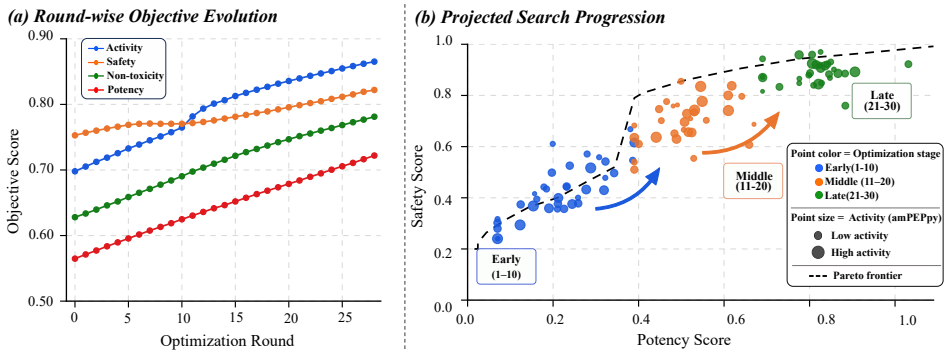

Figure 3: Impact of long-term memory on optimization dynamics. (a) Round-wise evolution and stage-wise means. (b) Projected search progression across three stages.

Contribution of Long-Term Memory and Memory Router. Figure 4(a) shows that full memory achieves a mean HV of 0.820, compared with 0.547 for no memory and 0.738 for random memory. It outperforms random memory on all 11 objective subsets and no memory on 10, with pronounced gains on AHT, HTP, and $AHTP$. These results support the value of retrieving relevant experience for coordinating competing objectives. Figure 4(b) further shows that adaptive routing improves HV most strongly on potency-related subsets, particularly AP, ATP, and AHP, while its effect on HT is negligible. **These findings suggest complementary benefits from accumulating design experience and adapting its retrieval to the optimization state.** Detailed memory ablations and baseline comparisons are provided in Appendices I and J, respectively.

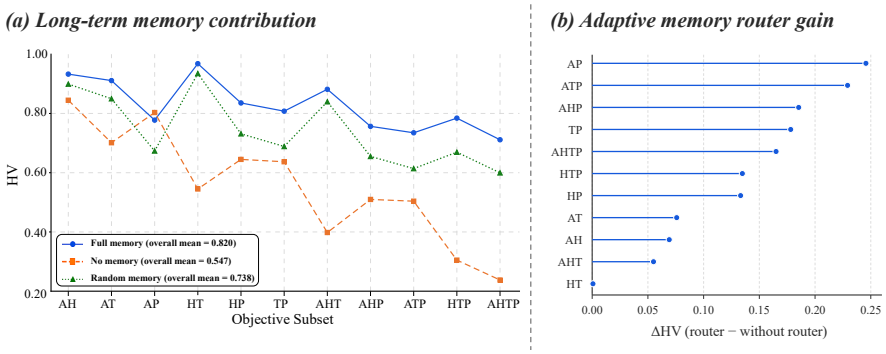

Figure 4: Contributions of long-term memory and memory router. (a) HV comparison among full, no, and random memory. (b) HV gains from adaptive routing relative to retrieval without the router.

5.4 Cross-Task Transferability of Design Experience (RQ3)

To examine whether design experience can transfer across pathogen groups, we transfer G^- experience to three G^+ pathogens. Table 2 compares five settings: empty memory ($M_\varnothing$), self experience (M_S), G^- experience (M^{G^-}), their combination ($M_S^{G^-}$), and the combined setting without MIC information ($M_S^{G^-\setminus\mathrm{MIC}}$). Across the 15 pathogen–objective comparisons, M^{G^-} outperforms $M_\varnothing$ in 14 cases, while $M_S^{G^-\setminus\mathrm{MIC}}$ outperforms $M_\varnothing$ in all 15 cases. This suggests that transferred experience can provide useful activity and safety information. For the *E. faecium* task, $M_S^{G^-}$ increases HV_{HTP} from 0.5524 to 0.9186 and HV_{AHTP} from 0.4017 to 0.7362 relative to M_S, providing a case-specific example of how transferred experience may complement target-specific knowledge. **Overall, although the gains vary across target pathogens and memory settings, these results suggest the transferability of AMP design experience across pathogen groups.**

Table 2: HV under five memory settings for three G(+) pathogens. Best values are bold.

Memory	*E. faecalis* (G+)					*E. faecium* (G+)					*S. aureus* (G+)				
	AH	AT	AHT	HTP	AHTP	AH	AT	AHT	HTP	AHTP	AH	AT	AHT	HTP	AHTP
$M_{\varnothing}$	0.6962	0.8316	0.6847	0.4449	0.3703	0.7318	0.8702	0.7133	0.5679	0.4934	0.7193	0.8612	0.7123	0.4643	0.3959
M_S	0.9684	0.9646	0.9543	**0.9513**	**0.9064**	0.7518	0.7391	0.7366	0.5524	0.4017	0.9125	0.9372	0.9097	**0.6933**	**0.6189**
M^{G^-}	0.9213	0.9073	0.8993	0.5805	0.4916	0.8884	0.8648	0.8555	0.8291	0.6481	0.9509	0.9331	0.9227	0.6713	0.5971
$M_S^{G^-}$	0.9164	0.8889	0.8797	0.5764	0.4988	0.8711	0.8570	0.8505	**0.9186**	**0.7362**	**0.9556**	**0.9376**	**0.9296**	0.6720	0.6100
$M_S^{G^-\setminus \mathrm{MIC}}$	**0.9845**	**0.9761**	**0.9593**	0.8358	0.8055	**0.8943**	**0.9164**	**0.8873**	0.7180	0.6479	0.9007	0.9079	0.8991	0.6267	0.5555

5.5 Biological Plausibility and Structural Reliability (RQ4)

Figure 5 shows representative candidates for *E. coli*, *P. aeruginosa*, *K. pneumoniae*, and *S. aureus*. OmegaFold predicts predominantly α-helical structures in all four. Their mean pLDDT scores range from 78.6 to 89.8, indicating moderate to high confidence in the predicted local structures. Predominantly positive electrostatic surfaces and distinct hydrophobic and hydrophilic patches are consistent with amphipathic properties. All four candidates meet the predefined predictor criteria for antimicrobial activity, hemolytic safety, toxicity safety, and target-specific MIC. **These computational results support the structural plausibility and predicted feasibility of these candidates.**

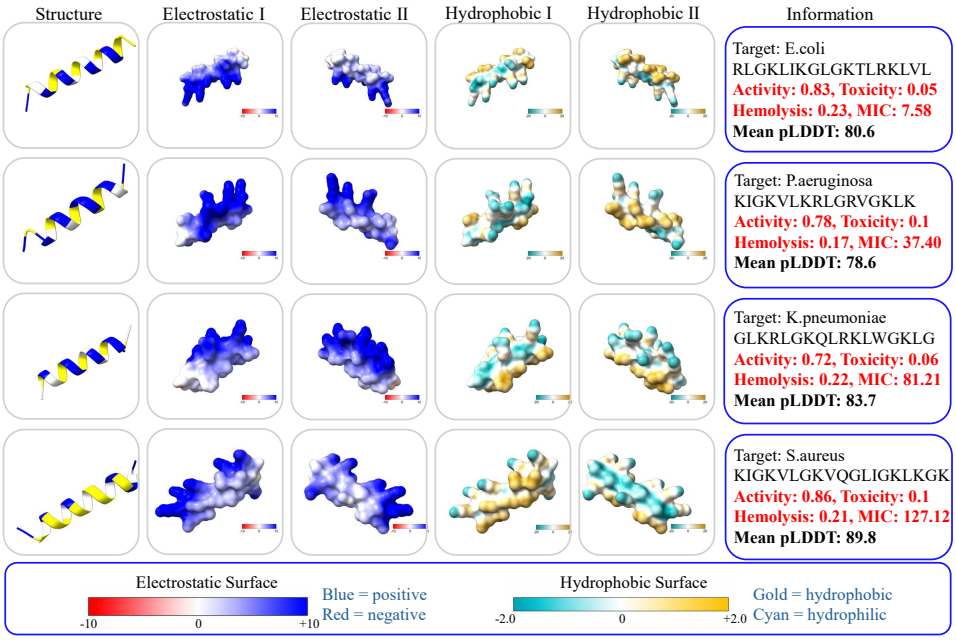

Figure 5: Structural and physicochemical profiles of four representative designed AMPs. Each row shows the predicted structure, two views of the electrostatic and hydrophobic surfaces, and the sequence, activity, toxicity, hemolysis, MIC, and mean pLDDT. Blue/red indicate positive/negative electrostatic potential. Gold/cyan indicate hydrophobic/hydrophilic surface regions.

6 Conclusion

PepMind formulates multi-objective AMP design as learning from reusable optimization experience. It converts evaluated successes and failures into structured memory and uses a learned, state-conditioned router to select the category and amount of experience for bottleneck-aware sequence refinement. Across seven pathogen-specific tasks, PepMind achieves strong multi-objective performance. Cross-pathogen experiments further show that experience accumulated for one pathogen can benefit another. Future work will experimentally validate the designed peptides, incorporate measured outcomes into memory, and extend PepMind to other peptide design tasks.

7 Ethics Statement

This work focuses on the computational design of antimicrobial peptides for therapeutic applications. All experiments are conducted computationally and do not involve human subjects, animal experiments, or experimental manipulation of pathogens. We recognize that generative methods for biological sequence design may have potential dual-use implications. However, PepMind is designed specifically to optimize antimicrobial activity, pathogen-specific potency, hemolytic safety, and toxicity safety, and does not involve the design or modification of pathogens, virulence factors, or mechanisms intended to increase pathogenicity. Generated peptides are computational candidates and require further experimental validation before any practical use.

8 AI Use Statement

During code development, AI tools were used to debugging and refinement of the experimental code. For scientific figures, generative AI tools were used to create or refine selected visual elements. AI tools were also used for language editing to improve clarity and readability. All AI-assisted outputs were reviewed and verified by the authors before being incorporated into this work.

9 Reproducibility Statement

We provide the methodological and experimental details required to reproduce this work in the main paper and appendix. Appendix A describes the predictors, evaluation tools, and large language models used in the experiments. Appendices C describes the memory router's training procedure, and Appendices D defines the memory relevance score. Appendix B specifies the objective thresholds, pathogen-specific MIC thresholds, and candidate feasibility criteria, while Appendix E provides the hypervolume evaluation protocol. Complete experimental results and additional validation are reported in Appendices F–J. Appendix K provides the main prompt templates used by PepMind. All PepMind experiments use DeepSeek-V4-Pro as the backbone LLM via API. All code will be made publicly available. Our code is available at https://anonymous.4open.science/r/PepMind-D16C/.

References

Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *International conference on machine learning*, pp. 11–20. PMLR, 2019.

Cezara Bucataru and Corina Ciobanasu. Antimicrobial peptides: Opportunities and challenges in overcoming resistance. *Microbiological research*, 286:127822, 2024.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

Axel Hollmann, Melina Martínez, Martín E Noguera, Marcelo T Augusto, Anibal Disalvo, Nuno C Santos, Liliana Semorile, and Paulo C Maffía. Role of amphipathicity and hydrophobicity in the balance between hemolysis and peptide–membrane interactions of three related antimicrobial peptides. *Colloids and Surfaces B: Biointerfaces*, 141:528–536, 2016.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Ryan Li, Yusuf Roohani, Lin Qiu, Shiyi Cao, Gavin Li, Junze Zhang, Di Yin, Rick Wierenga, Deniz Kavi, Sherry Liu, Tianwei She, Shruti Marwaha, Jennefer N. Carter, Xin Zhou, Matthew T. Wheeler, Jonathan A. Bernstein, Mengdi Wang, Peng He, Jingtian Zhou, Michael P. Snyder, Le Cong, Aviv Regev, and Jure Leskovec. Autonomous biomedical research with an artificial intelligence agent. *Science*, 393(6813):eadz4351, 2026. doi: 10.1126/science.adz4351. URL https://www.science.org/doi/abs/10.1126/science.adz4351.

Ruofan Jin, Zaixi Zhang, Mengdi Wang, and Le Cong. Stella: Self-evolving llm agent for biomedical research, 2025a. URL https://arxiv.org/abs/2507.02004.

Shuwen Jin, Zihan Zeng, Xiyan Xiong, Baicheng Huang, Li Tang, Hongsheng Wang, Xiao Ma, Xiaochun Tang, Guoqing Shao, Xingxu Huang, et al. Ampgen: an evolutionary information-reserved and diffusion-driven generative model for de novo design of antimicrobial peptides. *Communications Biology*, 8(1):839, 2025b.

Zitai Kong, Yifan Dong, Yixuan Wu, Zhaokang Liang, Jian Wu, and Hongxia Xu. Mp2d: Constrained monte carlo tree-guided diffusion for multi-objective protein sequence design. *arXiv preprint arXiv:2605.05829*, 2026.

Travis J Lawrence, Dana L Carper, Margaret K Spangler, Alyssa A Carrell, Tomás A Rush, Stephen J Minter, David J Weston, and Jessy L Labbé. ampeppy 1.0: a portable and accurate antimicrobial peptide prediction tool. *Bioinformatics*, 37(14):2058–2060, 2021.

Yanru Li, Xianghan Xu, Xiaohui Zhang, Zhihui Xu, Jiaqi Zhao, Ruiyu Zhu, Ziyu Wang, Wei Ran, Wenqian Zhao, Ningyang Yan, Yifan Leng, Zexu Miao, Xiaomin Wang, Liping Wang, Jinxin Liu, Cong Pian, and Jinhu Huang. Broadamp-gpt: Ai-driven generation of broad-spectrum antimicrobial peptides for combating multidrug-resistant eskape pathogens. *Gut microbes*, 17(1):2523811, 2025.

Yiping Liu, Xinyi Zhang, Yuansheng Liu, Yansen Su, Xiangxiang Zeng, and Gary G. Yen. Evolutionary multi-objective optimization in searching for various antimicrobial peptides [feature]. *IEEE Computational Intelligence Magazine*, 18(2):31–45, 2023. doi: 10.1109/MCI.2023.3245731.

Natalie Maus, Yimeng Zeng, Haydn Thomas Jones, Yining Huang, Gaurav Ng Goel, Alden Rose, Kyurae Kim, Hyun-Su Lee, Marcelo Der Torossian Torres, Fangping Wan, Cesar de la Fuente-Nunez, Mark Yatskar, Osbert Bastani, and Jacob R. Gardner. Purely agent-driven black-box optimization for biological design, 2026. URL https://arxiv.org/abs/2601.22382.

Ki Ha Min, Koung Hee Kim, Mi-Ran Ki, and Seung Pil Pack. Antimicrobial peptides and their biomedical applications: A review. *Antibiotics*, 13(9):794, 2024.

Yuki Murakami, Shoichi Ishida, Yosuke Demizu, and Kei Terayama. Design of antimicrobial peptides containing non-proteinogenic amino acids using multi-objective bayesian optimisation. *Digital Discovery*, 2(5):1347–1353, 2023.

Christopher J L Murray, Kevin Shunji Ikuta, Fablina Sharara, Lucien Swetschinski, Gisela Robles Aguilar, Authia Gray, Chieh Han, Catherine Bisignano, Puja Rao, Eve Wool, Sarah C Johnson, Annie J Browne, Michael Give Chipeta, Frederick Fell, Sean Hackett, Georgina Haines-Woodhouse, Bahar H Kashef Hamadani, Emmanuelle A P Kumaran, Barney McManigal, Sureeruk Achalapong, Ramesh Agarwal, Samuel Akech, Samuel Albertson, John Amuasi, Jason Andrews, Aleskandr Aravkin, Elizabeth Ashley, François-Xavier Babin, Freddie Bailey, Stephen Baker, Buddha Basnyat, Adrie Bekker, Rose Bender, James A Berkley, Adhisivam Bethou, Julia Bielicki, Suppawat Boonkasidecha, James Bukosia, Cristina Carvalheiro, Carlos Castañeda-Orjuela, Vilada Chansamouth, Suman Chaurasia, Sara Chiurchiù, Fazle Chowdhury, Rafai Clotaire Donatien, Aislinn J Cook, Ben Cooper, Tim R Cressey, Elia Criollo-Mora, Matthew Cunningham, Saffiatou Darboe, Nicholas P J Day, Maia De Luca, Klara Dokova, Angela Dramowski, Susanna J Dunachie, Thuy Duong Bich, Tim Eckmanns, Daniel Eibach, Amir Emami, Nicholas Feasey, Natasha Fisher-Pearson, Karen Forrest, Coralith Garcia, Denise Garrett, Petra Gastmeier, Ababi Zergaw Giref, Rachel Claire Greer, Vikas Gupta, Sebastian Haller, Andrea Haselbeck, Simon I Hay, Marianne Holm, Susan Hopkins, Yingfen Hsia, Kenneth C Iregbu, Jan Jacobs, Daniel Jarovsky, Fatemeh Javanmardi, Adam W J Jenney, Meera Khorana, Suwimon Khusuwan, Niranjan Kissoon, Elsa Kobeissi, Tomislav Kostyanev, Fiorella Krapp, Ralf Krumkamp, Ajay Kumar, Hmwe Hmwe Kyu, Cherry Lim, Kruy Lim, Direk Limmathurotsakul, Michael James Loftus, Miles Lunn, Jianing Ma, Anand Manoharan, Florian Marks, Jürgen May, Mayfong Mayxay, Neema Mturi, Tatiana Munera-Huertas, Patrick Musicha, Lilian A Musila, Marisa Marcia Mussi-Pinhata, Ravi Narayan Naidu, Tomoka Nakamura, Ruchi Nanavati, Sushma Nangia, Paul Newton, Chanpheaktra Ngoun, Amanda Novotney, Davis Nwakanma, Christina W Obiero, Theresa J Ochoa, Antonio Olivas-Martinez, Piero Olliaro, Ednah Ooko, Edgar Ortiz-Brizuela, Pradthana Ounchanum, Gideok D Pak, Jose Luis Paredes, Anton Yariv Peleg, Carlo Perrone, Thong Phe, Koukeo Phommasone, Nishad Plakkal, Alfredo Ponce de Leon, Mathieu

Raad, Tanusha Ramdin, Sayaphet Rattanavong, Amy Riddell, Tamalee Roberts, Julie Victoria Robotham, Anna Roca, Victor Daniel Rosenthal, Kristina E Rudd, Neal Russell, Helio S Sader, Weerawut Saengchan, Jesse Schnall, John Anthony Gerard Scott, Samroeng Seekaew, Mike Sharland, Madhusudhan Shivamallappa, Jose Sifuentes-Osornio, Andrew J Simpson, Nicolas Steenkeste, Andrew James Stewardson, Temenuga Stoeva, Nidanuch Tasak, Areerat Thaiprakong, Guy Thwaites, Caroline Tigoi, Claudia Turner, Paul Turner, H Rogier van Doorn, Sithembiso Velaphi, Avina Vongpradith, Manivanh Vongsouvath, Huong Vu, Timothy Walsh, Judd L Walson, Seymour Waner, Tri Wangrangsimakul, Prapass Wannapinij, Teresa Wozniak, Tracey E M W Young Sharma, Kalvin C Yu, Peng Zheng, Benn Sartorius, Alan D Lopez, Andy Stergachis, Catrin Moore, Christiane Dolecek, and Mohsen Naghavi. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. *Lancet (London, England)*, 399(10325):629, 2022.

Mohsen Naghavi, Stein Emil Vollset, Kevin S Ikuta, Lucien R Swetschinski, Authia P Gray, Eve E Wool, Gisela Robles Aguilar, Tomislav Mestrovic, Georgia Smith, Chieh Han, et al. Global burden of bacterial antimicrobial resistance 1990–2021: a systematic analysis with forecasts to 2050. *The Lancet*, 404(10459):1199–1226, 2024.

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.

Anand Singh Rathore, Shubham Choudhury, Akanksha Arora, Purva Tijare, and Gajendra PS Raghava. Toxinpred 3.0: An improved method for predicting the toxicity of peptides. *Computers in biology and medicine*, 179:108926, 2024.

Anand Singh Rathore, Nishant Kumar, Shubham Choudhury, Naman Kumar Mehta, and Gajendra PS Raghava. Prediction of hemolytic peptides and their hemolytic concentration. *Communications Biology*, 8(1):176, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7584–7600, 2024.

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.

Marcelo DT Torres, Fangping Wan, and Cesar de la Fuente-Nunez. Deep learning reveals antibiotics in the archaeal proteome. *Nature microbiology*, 10(9):2153–2167, 2025.

Marcelo DT Torres, Yimeng Zeng, Fangping Wan, Natalie Maus, Jacob Gardner, and Cesar de la Fuente-Nunez. A generative artificial intelligence approach for peptide antibiotic optimization. *Nature Machine Intelligence*, pp. 1–16, 2026.

Andrejs Tucs, Duy Phuoc Tran, Akiko Yumoto, Yoshihiro Ito, Takanori Uzawa, and Koji Tsuda. Generating ampicillin-level antimicrobial peptides with activity-aware generative adversarial networks. *ACS omega*, 5(36):22847–22851, 2020.

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.

Fangping Wan, Felix Wong, James J Collins, and Cesar de la Fuente-Nunez. Machine learning for antimicrobial peptide identification and design. *Nature Reviews Bioengineering*, 2(5):392–407, 2024.

Beilun Wang, Peijun Lin, Yuwei Zhong, Xiao Tan, Yangyang Shen, Yi Huang, Kai Jin, Yan Zhang, Ying Zhan, Dian Shen, Meng Wang, Zhou Yu, and Yihan Wu. Explainable deep learning and virtual evolution identifies antimicrobial peptides with activity against multidrug-resistant human pathogens. *Nature microbiology*, 10(2):332–347, 2025a.

Guangshun Wang, Cindy Schmidt, Xia Li, and Zhe Wang. Apd6: the antimicrobial peptide database is expanded to promote research and development by deploying an unprecedented information pipeline. *Nucleic Acids Research*, 54(D1):D363–D374, 2026.

Jen-Hung Wang and Ting-Yi Sung. Toxteller: predicting peptide toxicity using four different machine learning approaches. *ACS omega*, 9(29):32116–32123, 2024.

Jike Wang, Jianwen Feng, Yu Kang, Peichen Pan, Jingxuan Ge, Yan Wang, Mingyang Wang, Zhenxing Wu, Xingcai Zhang, Jiameng Yu, Xujun Zhang, Tianyue Wang, Lirong Wen, Guangning Yan, Yafeng Deng, Hui Shi, Chang-Yu Hsieh, Zhihui Jiang, and Tingjun Hou. Discovery of antimicrobial peptides with notable antibacterial potency by an llm-based foundation model. *Science advances*, 11(10):eads8932, 2025b.

Li Wang, Xiangzheng Fu, Jiahao Yang, Xinyi Zhang, Xiucai Ye, Tetsuya Sakurai, Xiangxiang Zeng, and Yiping Liu. Moformer: navigating the antimicrobial peptide design space with pareto-based multi-objective transformer. *Briefings in Bioinformatics*, 26(6):bbaf376, 2025c.

Li Wang, Yiping Liu, Xiangzheng Fu, Xiucai Ye, Junfeng Shi, Gary G Yen, Quan Zou, Xiangxiang Zeng, and Dongsheng Cao. Hmamp: designing highly potent antimicrobial peptides using a hypervolume-driven multiobjective deep generative model. *Journal of Medicinal Chemistry*, 68 (8):8346–8360, 2025d.

Rui Wang, Tao Wang, Linlin Zhuo, Jinhang Wei, Xiangzheng Fu, Quan Zou, and Xiaojun Yao. Diff-amp: tailored designed antimicrobial peptide framework with all-in-one generation, identification, prediction and optimization. *Briefings in Bioinformatics*, 25(2):bbae078, 2024.

Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022.

Peilin Xie, Lantian Yao, Jiahui Guan, Chia-Ru Chung, Zhihao Zhao, Feiyu Long, Zhenglong Sun, Tzong-Yi Lee, and Ying-Chih Chiang. Consamphemo: A computational framework for predicting hemolysis of antimicrobial peptides based on machine learning approaches. *Protein Science*, 34 (7):e70087, 2025.

Junhao Xiong, Ishan Gaur, Maria Lukarska, Hunter Nisonoff, Luke M Oltrogge, David F Savage, and Jennifer Listgarten. Property guidance for protein sequence generative models with protein-guide. *Nature Biotechnology*, pp. 1–13, 2026.

Lantian Yao, Jiahui Guan, Peilin Xie, Chia-Ru Chung, Zhihao Zhao, Danhong Dong, Yilin Guo, Wenyang Zhang, Junyang Deng, Yuxuan Pang, Yulan Liu, Yunlu Peng, Jorng-Tzong Horng, Ying-Chih Chiang, and Tzong-Yi Lee. dbamp 3.0: updated resource of antimicrobial activity and structural annotation of peptides in the post-pandemic era. *Nucleic acids research*, 53(D1):D364–D376, 2025.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=iW1zfncsbc`.

A Details of predictors and tools

amPEPpy. amPEPpy is a sequence-based predictor for identifying antimicrobial peptides (Lawrence et al., 2021). It uses a random forest classifier. Each input sequence is encoded by 105 distribution descriptors. These descriptors summarize the positions of amino acids with different physicochemical properties along the sequence. The input is a peptide sequence composed of standard amino acids. The model outputs an AMP or non-AMP prediction and a score for the AMP class. We use this score as the antimicrobial activity objective $A(x)$. A higher score indicates a higher predicted probability of antimicrobial activity.

HemoPI2. HemoPI2 predicts the hemolytic activity of peptides against mammalian red blood cells (Rathore et al., 2025). It provides both classification and regression models. The classification module includes random forest models, ESM2-t6-based models, and hybrid models that combine sequence prediction with MERCI motif matching. The regression module uses a random forest regressor to estimate the concentration required to cause 50% hemolysis, such as the HC_{50} or EC_{50}. Both modules take the primary peptide sequence as input. In this study, we use the classification output. The model returns a hemolytic score between 0 and 1, where a higher score indicates a higher predicted risk of hemolysis. We convert this score into the hemolytic safety objective $H(x)$, so that a higher value represents greater predicted safety.

ToxinPred3. ToxinPred3 is a sequence-based tool for classifying peptides as toxic or non-toxic (Rathore et al., 2024). Its machine learning model uses amino acid composition and dipeptide composition as input features. The tool also provides a hybrid model that combines an Extra Trees classifier with toxic motifs identified by MERCI. The input is a peptide sequence in FASTA or single-letter format. The output includes a toxic or non-toxic label and a toxicity score between 0 and 1. A higher score indicates a higher predicted toxic potential. We reverse this score to obtain the toxicity safety objective $T(x)$. Thus, a higher value of $T(x)$ represents lower predicted toxicity.

APEX. APEX is a multitask deep learning model for predicting pathogen-specific antimicrobial activity (Torres et al., 2025). It takes a peptide sequence as input and does not require structural information. Its sequence encoder combines recurrent and attention-based neural networks. The encoder produces a shared sequence representation. This representation is passed to two prediction heads. One head performs binary AMP classification using public peptide data. The other head uses a fully connected neural network to predict antimicrobial activity against specific bacterial strains. The final APEX model combines predictions from an ensemble of selected models. In this study, we use the pathogen-specific regression output to obtain the predicted MIC. When several strains are available for one pathogen, their predicted MIC values are averaged. The resulting value is used to compute the target-specific potency objective $P_s(x)$.

OmegaFold v1. OmegaFold v1 is a single-sequence model for protein and peptide structure prediction (Wu et al., 2022). It predicts a three-dimensional structure directly from the primary amino acid sequence. It does not require multiple sequence alignments or structural templates. OmegaFold contains three main parts. OmegaPLM first extracts residue-level and residue-pair representations from the input sequence. Geoformer then updates these representations using geometric relations between residues. A structure module finally converts the representations into three-dimensional atomic coordinates. For each input sequence, OmegaFold produces a structure file in PDB format. It also provides a confidence value for each residue in the B-factor field. We use the mean pLDDT score to measure structural reliability. OmegaFold is used only for auxiliary screening and does not affect the four primary optimization objectives.

AMP-CLIP. AMP-CLIP is a deep learning model for classifying peptides as AMPs or non-AMPs (Wang et al., 2025a). It contains four sequence models based on CNN, LSTM, Attention, and Transformer architectures. The input is a peptide sequence in FASTA format. Each amino acid is encoded by a 20-dimensional one-hot vector, and sequences are padded to a fixed length. Each model outputs an AMP probability. AMP-CLIP uses a strict ensemble rule. A peptide is classified as an AMP only when all four models predict the AMP class. In this study, AMP-CLIP is used as an external predictor of antimicrobial activity.

AMP-READ. AMP-READ is a sequence-based regression model for predicting antimicrobial potency (Wang et al., 2025a). It uses the same four model types as AMP-CLIP: CNN, LSTM, Attention, and Transformer. The regression models are trained after the AMP classification stage. This allows them to reuse sequence features learned from AMP identification. The input is a peptide sequence in FASTA format. Each model produces a continuous MIC prediction. The four predictions are then combined using learned linear weights. The released implementation reports both the ensemble result and the output of each model. A lower predicted MIC indicates stronger antimicrobial potency. We use the ensemble output as an external measure of AMP potency.

ToxTeller. ToxTeller is a sequence-based tool for peptide toxicity prediction (Wang & Sung, 2024). It provides four classifiers based on logistic regression, support vector machine, random forest, and XGBoost. Each peptide is represented by sequence composition and physicochemical features. The sequence features include amino acid composition, dipeptide composition, and one-hot encodings of the first and last eight residues. The physicochemical features include the mean values of 13 residue properties, such as hydrophobicity, volume, charge, and isoelectric point. The input is a peptide sequence in FASTA format. Each classifier outputs a binary toxicity prediction, where 1 represents a toxic peptide and 0 represents a non-toxic peptide. In this study, we use the output of the logistic regression classifier as the external toxicity prediction.

ConsAMP-Hemo. ConsAMP-Hemo is a two-stage framework for predicting peptide hemolysis (Xie et al., 2025). The first stage performs binary classification of hemolytic and non-hemolytic peptides. It combines ProtBert-BFD sequence representations with sequence-derived features. Contrastive learning is used to separate peptides from the two hemolytic classes. The classification model outputs a class label and a score for each class. The second stage uses hand-crafted physicochemical features and an XGBoost regressor to predict HC_{50}. The input to both stages is the primary peptide sequence. The classification output indicates whether the peptide is likely to cause hemolysis. The regression output estimates the concentration that causes 50% hemolysis. In this study, we use the classification stage as an external measure of hemolytic safety.

Large language models. We use DeepSeek-V4-Pro as the LLM for all PepMind components throughout the experiments. GPT-5.6-Sol, GLM-5.2, Qwen-3.5, and Kimi-2.6 are used only for external evaluation.

B Objective Thresholds and Candidate Feasibility

We use fixed thresholds to determine whether a candidate satisfies the four primary design objectives. All thresholds are set before optimization and remain unchanged across methods. The four objectives are defined in the same direction, where a higher value indicates a better candidate.

Antimicrobial activity threshold. The antimicrobial activity objective is the AMP score predicted by amPEPpy:

$$A(x) = p_{\mathrm{AMP}}(x). \tag{10}$$

A candidate satisfies the activity requirement when

$$A(x) \geq \tau_A, \qquad \tau_A = 0.7. \tag{11}$$

The threshold is set above a neutral classification boundary. It retains candidates with a clear antimicrobial signal and removes sequences with uncertain AMP predictions.

Hemolytic safety threshold. Let $p_{\mathrm{hemo}}(x)$ denote the hemolytic score produced by HemoPI2. A higher raw score indicates a higher predicted risk of hemolysis. We reverse this score to define the hemolytic safety objective:

$$H(x) = 1 - p_{\mathrm{hemo}}(x). \tag{12}$$

A candidate passes the hemolytic safety filter when

$$p_{\mathrm{hemo}}(x) \leq 0.4, \tag{13}$$

which is equivalent to

$$H(x) \geq \tau_H, \qquad \tau_H = 0.6. \tag{14}$$

This threshold removes candidates with a high predicted risk of red blood cell damage. It also provides a stricter safety requirement than using an uncertain score near the middle of the prediction range.

Toxicity safety threshold. Let $p_{\mathrm{tox}}(x)$ denote the toxicity score produced by ToxinPred3. A higher score indicates a higher predicted toxic potential. We define the toxicity safety objective as

$$T(x) = 1 - p_{\mathrm{tox}}(x). \tag{15}$$

A candidate satisfies the toxicity requirement when

$$p_{\mathrm{tox}}(x) \leq 0.4, \tag{16}$$

or equivalently,

$$T(x) \geq \tau_T, \qquad \tau_T = 0.6. \tag{17}$$

The cutoff provides a common safety filter and excludes candidates with high predicted toxicity. The same threshold is used for all pathogens because this prediction is peptide-specific rather than pathogen-specific.

Target-specific potency threshold. For target pathogen s, APEX predicts the MIC of candidate x. When several strain models are available for the same pathogen, we calculate the mean predicted MIC:

$$\overline{\mathrm{MIC}}_s(x) = \frac{1}{J_s} \sum_{j=1}^{J_s} \widehat{\mathrm{MIC}}_{s,j}(x), \tag{18}$$

where J_s is the number of available strain models for pathogen s. We then convert the predicted MIC into a direction-aligned potency score:

$$P_s(x) = \left(1 + \frac{\overline{\mathrm{MIC}}_s(x)}{\mathrm{MIC}_s^{\mathrm{thr}}}\right)^{-1}. \tag{19}$$

A candidate satisfies the potency requirement when

$$P_s(x) \geq \tau_P, \qquad \tau_P = 0.5. \tag{20}$$

This condition is equivalent to

$$\overline{\mathrm{MIC}}_s(x) \leq \mathrm{MIC}_s^{\mathrm{thr}}. \tag{21}$$

We set pathogen-specific MIC thresholds at three levels: 64 μM, 128 μM, and 256 μM. The levels follow a two-fold increase from the base value of 64 μM. Their assignment is based on the MIC distribution in the original APEX training data and our preliminary experimental results. Table 3 shows the training-data statistics used for this assignment.

Table 3: Pathogen-specific MIC statistics derived from the training data provided in the original APEX study.

Pathogen	APEX Mean MIC	APEX Min. MIC	Gram
E. coli	116–225	18	G−
A. baumannii	134	18	G−
E. faecium	123	6	G+
P. aeruginosa	208–254	41	G−
K. pneumoniae	381	150	G−
S. aureus	419–436	88	G+
E. faecalis	471	230	G+

Pathogens with lower mean MIC values and low observed minima are assigned the strict threshold of 64 μM. This group includes *E. coli*, *A. baumannii*, and *E. faecium*. *P. aeruginosa* has a higher MIC distribution and is assigned the intermediate threshold of 128 μM. Pathogens with the highest mean MIC values are assigned the relaxed threshold of 256 μM. This group includes *K. pneumoniae*, *S.*

aureus, and *E. faecalis*. The Gram type is reported only as background information. It is not used as the direct rule for threshold assignment.

These thresholds account for the different MIC ranges across pathogen-specific APEX models. They also avoid using one cutoff that would make some tasks much harder than others. The thresholds are used only for computational candidate screening. They should not be interpreted as clinical susceptibility breakpoints.

Table 4: Pathogen-specific MIC thresholds used for candidate evaluation.

Pathogen	Number of Strains	MIC Threshold
E. coli	3	64 μM
A. baumannii	1	64 μM
E. faecium	1	64 μM
P. aeruginosa	2	128 μM
K. pneumoniae	1	256 μM
S. aureus	2	256 μM
E. faecalis	1	256 μM

Overall feasibility. A candidate is feasible only when it satisfies all four requirements:

$$A(x) \geq 0.7, \qquad H(x) \geq 0.6, \qquad T(x) \geq 0.6, \qquad P_s(x) \geq 0.5. \tag{22}$$

Equivalently, the raw hemolysis and toxicity scores must both be no greater than 0.4, and the mean predicted MIC must not exceed the pathogen-specific threshold in Table 4. Sequence novelty and structural reliability are assessed after this primary feasibility check. They are not included in the four-objective feasibility label.

C Memory Router

Counterfactual training data construction. We construct the router training data from optimization trajectories using counterfactual replay. Each trajectory consists of multiple optimization rounds for a target pathogen. For round $r \geq 2$, the optimization state at the end of round $r-1$ is treated as a routing decision point. Therefore, both the state representation and the available memory contain only information observed before round r.

Specifically, the feature vector follows the fixed ordering

$$\mathbf{h}_r = [\mathbf{f}_{r-1}, \mathbf{d}_{r-1}, \Delta\mathbf{f}_{r-1}, p_r, \gamma_{r-1}, \ell_{r-1}, \xi_{r-1}] \in \mathbb{R}^{16}, \tag{23}$$

where $\mathbf{f}_{r-1} \in \mathbb{R}^4$ contains the current activity, hemolytic-safety, toxicity-safety, and potency summaries; $\mathbf{d}_{r-1} = \mathbb{I}(\mathbf{f}_{r-1} < \boldsymbol{\tau})$ contains their violation indicators; and $\Delta\mathbf{f}_{r-1} = \mathbf{f}_{r-1} - \mathbf{f}_{r-2}$ contains their round-to-round changes. The remaining four entries are the normalized round progress p_r, feasible-candidate ratio γ_{r-1}, stagnation measure ℓ_{r-1}, and population diversity ξ_{r-1}, respectively. All continuous features are standardized using statistics estimated from the training set only.

Each decision state is represented by a 16-dimensional feature vector $\mathbf{h}_r$. It summarizes the current optimization status using the four predictor outputs for activity, hemolytic safety, toxicity safety, and target-specific potency, together with their threshold-violation indicators. The remaining features describe round-to-round changes, optimization progress, candidate feasibility, stagnation, and population diversity. The associated memory pool contains only records accumulated in rounds $1, \ldots, r-1$, which prevents future information from entering the routing decision.

For each state $\mathbf{h}_r$, we evaluate a set of alternative routing actions. The action space is defined as

$$\mathcal{A} = \{\text{none}\} \cup (\mathcal{M} \times \mathcal{N}), \tag{24}$$

where

$$\mathcal{M} = \{\text{success}, \text{activity-failure}, \text{hemolysis-failure}, \text{toxicity-failure}, \text{potency-failure}\}, \tag{25}$$

and $\mathcal{N} = \{1, 2, 4, 8\}$ denotes the candidate retrieval sizes. The `none` action represents retrieval without memory and is included only once in the action space. Actions requiring a memory category that is not yet available are excluded at the corresponding decision point.

For each valid action $a \in \mathcal{A}$, we replay the subsequent candidate refinement from the same historical state while changing only the routing decision. The selected memory is used to guide strategy planning and peptide generation, after which the generated candidates are evaluated by the same four predictors used in the main optimization procedure. Each replay is isolated from the original trajectory so that counterfactual evaluation does not modify the historical memory.

We quantify the benefit of action a by its improvement in four-objective hypervolume:

$$\Delta\mathrm{HV}_a = \mathrm{HV}(\mathcal{D}_{r-1} \cup \mathcal{C}_a) - \mathrm{HV}(\mathcal{D}_{r-1}), \tag{26}$$

where $\mathcal{D}_{r-1}$ is the archive available before round r and $\mathcal{C}_a$ is the candidate set obtained under action a. The four objectives are normalized to $[0, 1]$. To balance optimization gain and retrieval cost, we define

$$u_a = \Delta\mathrm{HV}_a - \lambda\, c(a), \qquad \lambda = 0.002, \tag{27}$$

where $c(a) = n$ for a memory action with retrieval size n, and $c(a) = 0$ for the no-memory action.

The utilities are converted into soft supervision rather than selecting only the highest-utility action:

$$q(a \mid \mathbf{h}_r) = \frac{\exp(u_a/\tau)}{\sum_{a' \in \mathcal{A}_r} \exp(u_{a'}/\tau)}, \qquad \tau = 1.0, \tag{28}$$

where $\mathcal{A}_r$ denotes the valid action set at decision point r. The resulting joint distribution is marginalized to produce the six-class memory-type target and the four-class retrieval-size target. The former includes the no-memory decision, whereas the latter is defined over memory-using actions only. In this way, the router learns not only the highest-utility action, but also the relative preference among alternative retrieval strategies.

The resulting dataset contains 110 routing decision states from four target pathogens. We use 99 states for training and 11 for validation. Splitting is performed at the trajectory level rather than at individual decision points to avoid sharing closely related optimization states across the two sets.

Router architecture and optimization. The router is implemented as a lightweight dual-head multilayer perceptron. Given $\mathbf{h}_r \in \mathbb{R}^{16}$, a shared backbone with hidden dimensions 64 and 32 produces a compact state representation. Each hidden layer is followed by a ReLU activation. Two independent linear heads then predict the memory-type and retrieval-size distributions:

$$\hat{\mathbf{p}}_{\mathrm{type}} = \mathrm{softmax}(\mathbf{z}_{\mathrm{type}}), \qquad \hat{\mathbf{p}}_{\mathrm{size}} = \mathrm{softmax}(\mathbf{z}_{\mathrm{size}}). \tag{29}$$

Both heads are trained against the counterfactual soft targets using cross-entropy. Let $\mathbf{q}_{\mathrm{type}}$ and $\mathbf{q}_{\mathrm{size}}$ denote the corresponding target distributions. The training objective is

$$\mathcal{L}_{\mathrm{router}} = \mathcal{L}_{\mathrm{type}} + w_r \mathcal{L}_{\mathrm{size}}, \tag{30}$$

with

$$\mathcal{L}_{\mathrm{type}} = -\sum_m q_m \log \hat{p}_m, \qquad \mathcal{L}_{\mathrm{size}} = -\sum_n q_n \log \hat{p}_n, \tag{31}$$

and

$$w_r = 1 - q_{\mathrm{none}}. \tag{32}$$

The weighting term reflects the fact that retrieval size is relevant only when memory is used. Thus, decision states that strongly favor the no-memory action contribute less to the size-head objective.

Input features are standardized using statistics estimated from the training set only. We optimize the router with Adam using a learning rate of 10^{-3} for at most 300 epochs. The combined validation loss is evaluated every 25 epochs, and the checkpoint with the lowest validation loss is used for subsequent optimization experiments.

Learned routing behavior. The counterfactual training data show a clear preference for potency-failure memory. Accordingly, the learned type head assigns substantial probability mass to this category across many decision states. This behavior is consistent with the optimization trajectories, where improving target-specific potency frequently produces the largest marginal gain in four-objective hypervolume. In contrast, the size head remains more state-dependent and distributes its predictions across the four retrieval sizes. This suggests that the amount of useful historical context varies with the current optimization state even when the preferred memory category is relatively stable.

Ablation study. We evaluate the contribution of adaptive memory routing by comparing PepMind with the learned router against a variant using the original heuristic retrieval strategy. Both variants use the same initial memory and optimization settings. Experiments are conducted on four target pathogens, with 10 independent runs for each setting.

As shown in Fig. 6a, introducing the router improves overall hypervolume across all 11 objective subsets, although the change for H+T is minimal. The largest absolute gains occur in combinations involving target-specific potency, particularly A+P, A+T+P, and A+H+P (Fig. 6b). For the full A+H+T+P objective set, the overall HV increases from 0.57668 to 0.74168, corresponding to a relative improvement of 28.6%. Pathogen-specific results show improvements in 43 of the 44 pathogen–subset combinations, with only a slight decrease for H+T in *E. faecalis*. The largest gains are observed for *E. faecium*, whose full-objective HV increases from 0.46190 to 0.71923, a relative improvement of 55.7% (Fig. 6c). These results suggest that adaptive memory routing supports multi-objective optimization, particularly when target-specific potency is included.

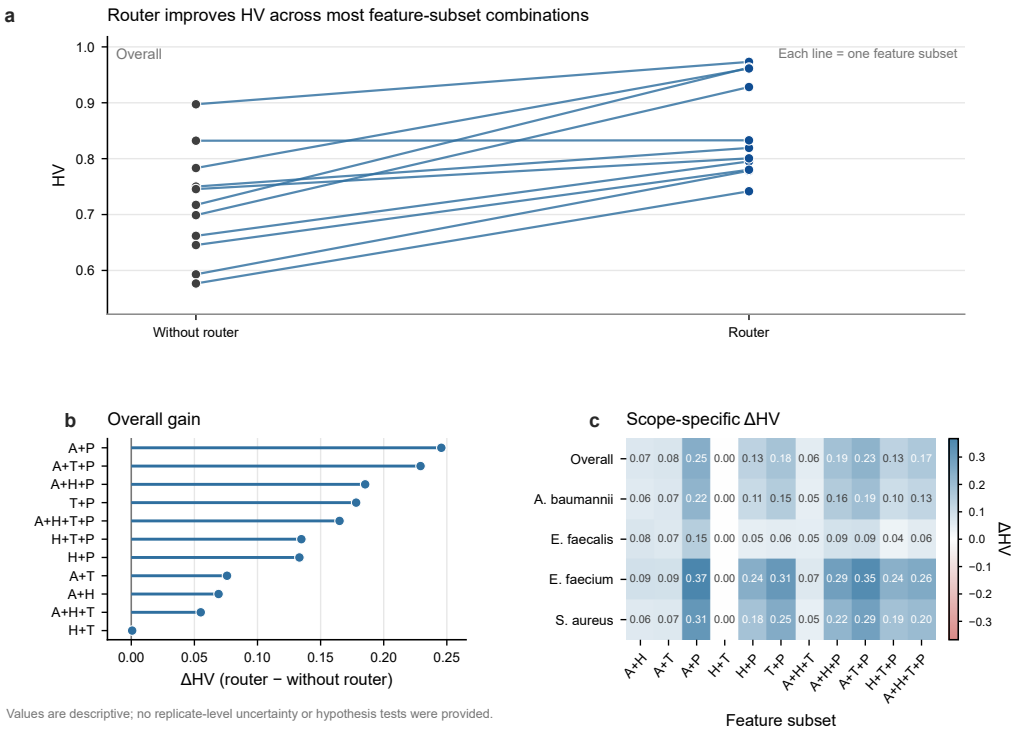

Figure 6: Ablation of the adaptive memory router. (a) Hypervolume comparison between PepMind with and without the learned router across different objective subsets. (b) Overall hypervolume improvement introduced by adaptive routing. (c) Pathogen-specific hypervolume changes across objective subsets.

D Definition of the Memory Relevance Score

For a candidate x and the category c_r selected by the router, the relevance of an eligible memory record m_i is scored as

$$\rho(m_i, x) = 0.8 S_{c_r}(m_i, x) + 0.2 S_{\text{phys}}(m_i, x), \tag{33}$$

where S_{c_r} is a category-specific priority score and S_{phys} is a physicochemical compatibility score. Both terms are bounded in $[0, 1]$, and eligible records are ranked in descending order of $\rho(m_i, x)$. The weighting makes the router-selected category the primary ranking criterion, with physicochemical compatibility serving as an auxiliary prior.

Category-specific priority score. Let a_i, h_i, t_i, and μ_i denote the predicted antimicrobial activity, hemolysis risk, toxicity risk, and target-specific MIC of the sequence stored in m_i, respectively. Let θ_s denote the MIC threshold for target pathogen s. We define

$$S_{c_r}(m_i, x) = \begin{cases} a_i[1 - \max(h_i, t_i)] \min\left(1, \dfrac{\theta_s}{\max(\mu_i, \theta_s)}\right), & c_r = \text{success}, \\ 1 - a_i, & c_r = \text{activity_failure}, \\ h_i, & c_r = \text{hemolysis_failure}, \\ t_i, & c_r = \text{toxicity_failure}, \\ \dfrac{\mu_i}{\mu_i + \theta_s}, & c_r = \text{potency_failure}. \end{cases} \tag{34}$$

For a failure category, records exhibiting a more severe failure in the corresponding objective receive a higher priority. Successful records are instead ranked by their joint antimicrobial activity, safety, and target-specific potency. In particular, the potency factor for successful records can be written as

$$\min\left(1, \frac{\theta_s}{\max(\mu_i, \theta_s)}\right) = \begin{cases} 1, & \mu_i \le \theta_s, \\ \theta_s / \mu_i, & \mu_i > \theta_s. \end{cases} \tag{35}$$

Therefore, records satisfying the MIC threshold receive the same maximum potency factor, whereas records exceeding the threshold are penalized monotonically. This prevents the retrieval score from unnecessarily favoring increasingly low MIC values after the required potency has been achieved.

Physicochemical compatibility score. The physicochemical score considers sequence length, charge density, and hydrophobic-residue ratio:

$$S_{\text{phys}}(m_i, x) = \frac{S_{\text{len}}(m_i, x) + S_{\text{charge}}(m_i) + S_{\text{hydro}}(m_i)}{3}, \tag{36}$$

where

$$S_{\text{len}}(m_i, x) = \max\left(0,\, 1 - \frac{|L_x - L_i|}{20}\right), \tag{37}$$

$$S_{\text{charge}}(m_i) = \max\left(0,\, 1 - \frac{|0.275 - q_i / L_i|}{0.3}\right), \tag{38}$$

$$S_{\text{hydro}}(m_i) = \max\left(0,\, 1 - \frac{|0.40 - \phi_i|}{0.3}\right). \tag{39}$$

Here, L_i is the length of the sequence in m_i, and

$$L_x = \frac{L_{\text{low}} + L_{\text{high}}}{2} \tag{40}$$

is the midpoint of the length range specified by the current query. The net charge is approximated as

$$q_i = n_i(\text{K}) + n_i(\text{R}) - n_i(\text{D}) - n_i(\text{E}), \tag{41}$$

without pH-dependent protonation or terminal-charge correction. The hydrophobic-residue ratio is

$$\phi_i = \frac{n_i(\text{A}, \text{V}, \text{I}, \text{L}, \text{M}, \text{F}, \text{W}, \text{Y})}{L_i}. \tag{42}$$

The length component is conditioned on the current query, whereas the reference charge density of 0.275 and hydrophobic-residue ratio of 0.40 are fixed AMP priors applied consistently across all tasks and experiments. Thus, S_{phys} provides a stable physicochemical compatibility prior rather than a fully query-dependent similarity measure. The category-specific term remains the dominant determinant of retrieval ranking.

E Hypervolume Evaluation

We use hypervolume (HV) to evaluate the quality of the non-dominated candidate set under different objective combinations. The four direction-aligned objectives are

$$\mathbf{f}_s(x) = [A(x),\, H(x),\, T(x),\, P_s(x)]\,, \tag{43}$$

where a larger value indicates a more desirable property.

All four objectives are naturally bounded within $[0, 1]$. The antimicrobial activity score $A(x)$ is obtained from amPEPpy. The hemolytic and toxicity safety scores are defined by reversing the corresponding risk scores:

$$H(x) = 1 - p_{\mathrm{hemo}}(x), \qquad T(x) = 1 - p_{\mathrm{tox}}(x). \tag{44}$$

The target-specific potency score is obtained by transforming the predicted MIC:

$$P_s(x) = \left(1 + \frac{\overline{\mathrm{MIC}}_s(x)}{\mathrm{MIC}_s^{\mathrm{thr}}}\right)^{-1}. \tag{45}$$

For the main comparison, because these objectives already share the same range and direction, no additional min–max normalization is applied. For ablation analyses, shared min–max bounds across all compared variants are used.

For an objective subset $\mathcal{S} \subseteq \{A, H, T, P\}$, let $\mathcal{P}_{s,\mathcal{S}}$ denote the non-dominated set of evaluated candidates projected onto the objectives in $\mathcal{S}$. Its hypervolume is defined as

$$\mathrm{HV}\left(\mathcal{P}_{s,\mathcal{S}}\right) = \lambda\left(\bigcup_{\mathbf{y}\in\mathcal{P}_{s,\mathcal{S}}} [\mathbf{r}_{\mathcal{S}}, \mathbf{y}]\right), \tag{46}$$

where $\lambda(\cdot)$ denotes the Lebesgue measure and $\mathbf{r}_{\mathcal{S}}$ is the reference point. We use

$$\mathbf{r}_{\mathcal{S}} = \mathbf{0}_{|\mathcal{S}|} \tag{47}$$

for every objective subset. Thus, the reference point is $(0, 0)$ for two-objective evaluation, $(0, 0, 0)$ for three-objective evaluation, and $(0, 0, 0, 0)$ for the full four-objective evaluation.

The same objective definitions and reference points are used for all methods and pathogen-specific tasks. A larger HV indicates that the non-dominated set contains candidates with stronger objective values and covers a wider range of trade-offs.

F Complete Multi-Objective Comparison across Seven Pathogens

Table 5 reports hypervolume results across seven pathogens and all 11 non-singleton objective subsets. All methods use the same generation budget within each task. PepMind achieves the best result in 46 of the 77 pathogen–objective-subset combinations. Its advantage is most consistent when efficacy and safety objectives are optimized together. It achieves the best AH and AHT results on all seven pathogens, and the best HT and HTP results on six pathogens. Under the complete AHTP setting, the variant ranks first on five pathogens. The sequential PepMind variant ranks first on the other two. Thus, a PepMind variant achieves the best full-objective result on all seven tasks.

The results are less consistent for AP, TP, and ATP. Some external generators perform better on these lower-dimensional subsets. For example, AMP-Designer shows strong activity–potency trade-offs under AP. However, this advantage does not remain stable after hemolytic and toxicity safety are added. PepMind performs more strongly on safety-inclusive subsets and the complete AHTP setting. These results show that PepMind is better at maintaining balanced efficacy and safety as more objectives are introduced.

Table 5: Complete hypervolume comparison across seven pathogen-specific AMP design tasks. Higher values indicate better multi-objective search quality, and the best result for each objective subset is shown in bold.

Method	AH	AT	AP	HT	HP	TP	AHT	AHP	ATP	HTP	AHTP
Task 1: *A. baumannii*											
PepMind	**0.6427**	**0.6716**	0.4962	**0.6998**	0.5170	0.5403	**0.5496**	0.4060	0.4243	**0.4421**	0.3472
PepMind (sequential)	0.5949	0.6642	0.6549	0.5565	**0.5487**	0.6126	0.4689	**0.4624**	**0.5162**	0.4325	**0.3645**
Biomni	0.4743	0.2258	0.0796	0.3184	0.1123	0.0535	0.1846	0.0651	0.0310	0.0437	0.0254
STELLA	0.5083	0.4778	0.3559	0.5393	0.4016	0.3775	0.3619	0.2695	0.2533	0.2859	0.1919
AMP-Designer	0.2178	0.5230	**0.7486**	0.1786	0.2556	**0.6138**	0.1427	0.2042	0.4902	0.1674	0.1337
BroadAMP-GPT	0.1915	0.3382	0.6040	0.1427	0.2549	0.4502	0.0961	0.1717	0.3033	0.1280	0.0862
PepGAN	0.2458	0.4330	0.7084	0.1788	0.2925	0.5153	0.1379	0.2257	0.3976	0.1641	0.1266
Diff-AMP	0.2207	0.1564	0.1705	0.1099	0.1198	0.0848	0.0616	0.0671	0.0476	0.0334	0.0187
Task 2: *P. aeruginosa*											
PepMind	**0.6430**	**0.6165**	0.4671	**0.6401**	**0.4850**	0.4650	**0.5037**	**0.3817**	0.3659	**0.3800**	**0.2990**
PepMind (sequential)	0.4921	0.5843	0.5463	0.4102	0.3835	0.4553	0.3434	0.3211	0.3812	0.2676	0.2241
Biomni	0.5079	0.3011	0.2023	0.3793	0.2548	0.1511	0.2408	0.1618	0.0959	0.1208	0.0767
STELLA	0.4915	0.5007	0.3823	0.5603	0.4278	0.4358	0.3713	0.2835	0.2888	0.3232	0.2142
AMP-Designer	0.3695	0.4559	**0.6596**	0.2994	0.4331	**0.5343**	0.2246	0.3249	**0.4009**	0.2632	0.1975
BroadAMP-GPT	0.1901	0.3947	0.5093	0.1655	0.2135	0.4433	0.1114	0.1438	0.2985	0.1252	0.0843
PepGAN	0.2629	0.5092	0.5826	0.2374	0.2717	0.5262	0.1783	0.2040	0.3951	0.1842	0.1383
Diff-AMP	0.2300	0.1442	0.1622	0.1389	0.1562	0.0979	0.0679	0.0763	0.0478	0.0461	0.0225
Task 3: *K. pneumoniae*											
PepMind	**0.6532**	0.5459	0.4562	0.5384	**0.4500**	0.3761	**0.4382**	**0.3662**	0.3061	**0.3018**	**0.2456**
PepMind (sequential)	0.4971	**0.6115**	0.4748	0.4519	0.3509	0.4317	0.3706	0.2878	0.3540	0.2617	0.2146
Biomni	0.4908	0.2794	0.2127	0.3796	0.2891	0.1645	0.2282	0.1737	0.0989	0.1344	0.0808
STELLA	0.4937	0.4778	0.2881	**0.5580**	0.3365	0.3256	0.3628	0.2188	0.2117	0.2473	0.1608
AMP-Designer	0.2759	0.4392	**0.7652**	0.1923	0.3350	**0.5332**	0.1526	0.2659	**0.4233**	0.1853	0.1471
BroadAMP-GPT	0.1831	0.2677	0.5939	0.1128	0.2502	0.3658	0.0744	0.1650	0.2411	0.1016	0.0670
PepGAN	0.2870	0.4170	0.6566	0.2146	0.3379	0.4909	0.1603	0.2524	0.3666	0.1886	0.1409
Diff-AMP	0.2122	0.0952	0.2183	0.0974	0.2233	0.1002	0.0444	0.1017	0.0456	0.0467	0.0213
Task 4: *S. aureus*											
PepMind	**0.6476**	**0.6872**	0.4649	0.6710	**0.4539**	0.4817	**0.5465**	**0.3697**	0.3923	**0.3830**	**0.3119**
PepMind (sequential)	0.5976	0.6741	0.3254	**0.6817**	0.3291	0.3712	0.5240	0.2530	0.2854	0.2886	0.2218
Biomni	0.4930	0.2722	0.2261	0.3671	0.3050	0.1684	0.2219	0.1844	0.1018	0.1373	0.0830
STELLA	0.4948	0.4828	0.2891	0.5609	0.3358	0.3277	0.3661	0.2192	0.2139	0.2485	0.1622
AMP-Designer	0.2264	0.4799	**0.7780**	0.1717	0.2784	**0.5900**	0.1366	0.2215	**0.4694**	0.1680	0.1336
BroadAMP-GPT	0.1828	0.2619	0.6281	0.1094	0.2624	0.3761	0.0724	0.1736	0.2487	0.1039	0.0687
PepGAN	0.2346	0.4669	0.7419	0.1820	0.2892	0.5755	0.1412	0.2244	0.4465	0.1741	0.1350
Diff-AMP	0.2194	0.1463	0.3104	0.1046	0.2220	0.1481	0.0580	0.1230	0.0820	0.0587	0.0325
Task 5: *E. faecalis*											
PepMind	**0.6551**	**0.6807**	0.4706	**0.6309**	0.4362	0.4532	**0.5304**	**0.3667**	0.3810	**0.3531**	**0.2969**
PepMind (sequential)	0.6260	0.5906	0.3083	0.5597	0.2922	0.2756	0.4549	0.2375	0.2240	0.2123	0.1725
Biomni	0.4848	0.2631	0.2020	0.3598	0.2763	0.1499	0.2142	0.1645	0.0893	0.1221	0.0727
STELLA	0.4951	0.4943	0.2372	0.5704	0.2737	0.2733	0.3736	0.1793	0.1790	0.2066	0.1353
AMP-Designer	0.3908	0.5242	**0.6864**	0.3590	**0.4701**	0.6306	0.2712	0.3551	0.4763	0.3262	0.2464
BroadAMP-GPT	0.2058	0.4550	0.5520	0.1877	0.2277	0.5036	0.1326	0.1608	0.3557	0.1467	0.1036
PepGAN	0.2412	0.6033	0.6597	0.2405	0.2629	**0.6578**	0.1870	0.2045	**0.5117**	0.2039	0.1586
Diff-AMP	0.1914	0.1728	0.2135	0.1567	0.1936	0.1748	0.0720	0.0889	0.0803	0.0728	0.0335
Task 6: *E. coli*											
PepMind	**0.6053**	**0.6355**	0.5256	**0.5920**	**0.4896**	**0.5141**	**0.4772**	**0.3947**	0.4144	**0.3860**	**0.3112**
PepMind (sequential)	0.3786	0.5693	0.6602	0.2844	0.3298	0.4960	0.2476	0.2871	**0.4317**	0.2157	0.1878
Biomni	0.4894	0.2849	0.1999	0.3794	0.2661	0.1550	0.2300	0.1613	0.0940	0.1251	0.0758
STELLA	0.4956	0.4687	0.3120	0.5430	0.3615	0.3419	0.3552	0.2364	0.2236	0.2590	0.1694
AMP-Designer	0.2608	0.3910	**0.7511**	0.1600	0.3073	0.4607	0.1277	0.2454	0.3678	0.1505	0.1202
BroadAMP-GPT	0.1826	0.1884	0.5825	0.0824	0.2547	0.2629	0.0532	0.1646	0.1699	0.0743	0.0480
PepGAN	0.2855	0.3479	0.6612	0.1695	0.3222	0.3927	0.1298	0.2466	0.3006	0.1465	0.1121
Diff-AMP	0.2106	0.1431	0.1499	0.1105	0.1157	0.0787	0.0577	0.0604	0.0411	0.0317	0.0166
Task 7: *E. faecium*											
PepMind	**0.6300**	0.6034	0.4779	**0.5932**	0.4699	0.4500	**0.4749**	0.3762	0.3602	0.3542	0.2835
PepMind (sequential)	0.6178	0.5927	0.6350	0.5371	**0.5755**	0.5520	0.4435	**0.4752**	0.4558	0.4131	**0.3411**
Biomni	0.4836	0.2535	0.2106	0.3490	0.2900	0.1520	0.2068	0.1719	0.0901	0.1240	0.0735
STELLA	0.4789	0.4939	0.2574	0.5675	0.2958	0.3050	0.3664	0.1909	0.1969	0.2263	0.1461
AMP-Designer	0.4421	0.6101	**0.7458**	0.4444	0.5432	0.7497	0.3462	0.4232	**0.5840**	**0.4254**	0.3314
BroadAMP-GPT	0.2076	0.6654	0.6347	0.2487	0.2372	**0.7603**	0.1854	0.1768	0.5666	0.2118	0.1578
PepGAN	0.3782	**0.6786**	0.6029	0.4304	0.3824	0.6861	0.3323	0.2953	0.5298	0.3360	0.2595
Diff-AMP	0.2998	0.4017	0.1923	0.3512	0.1681	0.2252	0.2057	0.0984	0.1319	0.1153	0.0675

G EXTERNAL VALIDATION COMPARISON

All peptides used for external evaluation were drawn from the candidate sets selected by the primary predictors. No additional candidate selection was performed using the external evaluators.

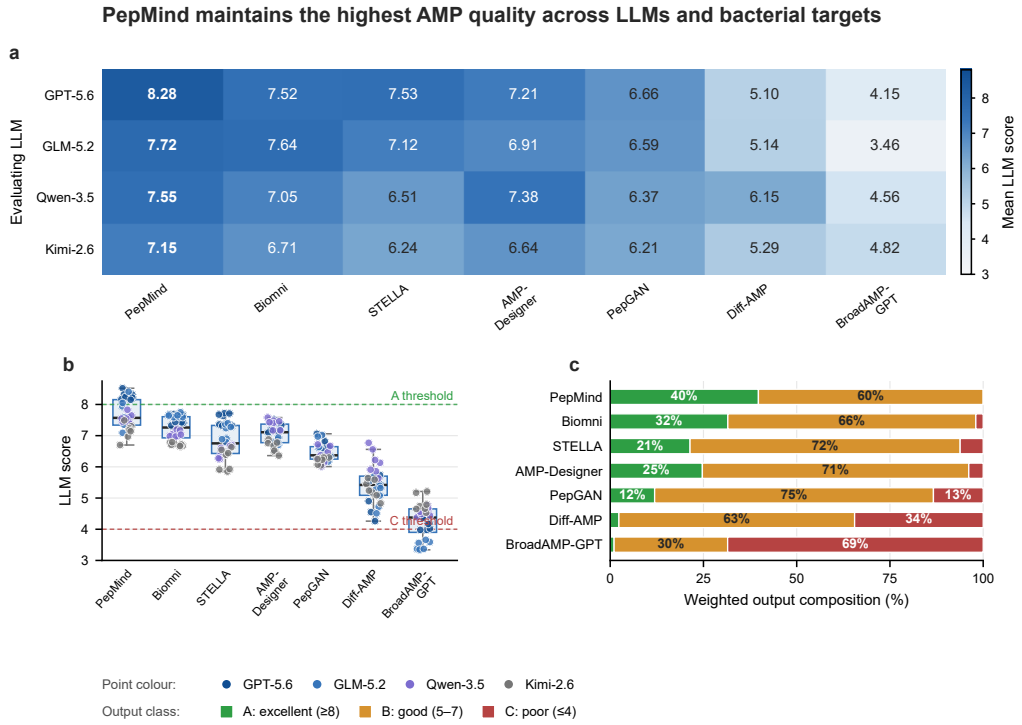

Figure 7: External LLM evaluation of generated AMP candidates. (a) Mean scores assigned by four LLM evaluators to candidates generated by each method. (b) Score distributions across LLM evaluators and bacterial targets; dashed lines indicate the thresholds for Class A (≥ 8) and Class C (≤ 4). (c) Weighted output composition of Class A (excellent), Class B (good), and Class C (poor) candidates.

G.1 LLM Baselines

To provide a model-independent reference for external validation, we further evaluate five general-purpose LLMs as AMP design baselines: DeepSeek-V4-Flash, Qwen-3.5, Kimi-2.6, GLM-5.2, and GPT-5.6. Candidate peptides generated by each model are assessed using the same activity, safety, and target-specific potency predictors adopted throughout PepMind.

As shown in Fig. 9, all five LLMs generate candidates with measurable antimicrobial and safety profiles, while showing distinct performance across models and target species. Their predicted MIC values also differ substantially, indicating different abilities to generate target-specific candidates (Fig. 9b). These differences are further reflected in both pairwise and higher-order hypervolume combinations (Fig. 9c).

These results show that general-purpose LLMs can provide non-trivial AMP design baselines under the same external evaluation protocol. We therefore use them as an additional model-independent reference for assessing PepMind, reducing the dependence of the comparison on any single generative model or evaluation perspective.

G.2 Results by other large language models

As shown in Figure 7(a), PepMind achieves the highest mean score under all four LLM evaluators. Figure 7(b) further shows consistently high scores across LLM evaluators and bacterial targets. Figure 7(c) groups the outputs into Class A (excellent, ≥ 8), Class B (good, 5–7), and Class C (poor, ≤ 4). PepMind produces 40% Class A and 60% Class B outputs, with no Class C outputs. Overall, these results indicate that PepMind maintains strong and consistent AMP quality across independent LLM evaluators and target pathogens.

The following prompt was used for all external LLM evaluators:

```
System:
You are an expert in antimicrobial peptide (AMP)
drug design. Evaluate the overall potential of each
peptide sequence as an antimicrobial drug candidate.

Evaluation criteria:

Antimicrobial activity:
Consider the content of positively charged residues
(K/R), hydrophobicity patterns, amphipathicity, and
known AMP motifs.

Safety:
Consider hemolytic risk and cytotoxicity. Excessive
hydrophobicity may indicate a higher hemolytic risk.

Structural plausibility:
Consider sequence length and amino acid composition.
A length of 8-35 amino acids is preferred.

Drug potential:
Consider the overall balance between antimicrobial
activity and safety.

User:
Peptide sequences:
<PEPTIDE_SEQUENCES>

Assign each sequence a score from 1 to 10.
A score of 10 indicates the best AMP candidate.

Return only one valid JSON object. Each key must be
the input index as a string, and each value must be
the corresponding score. Include all input indices
and do not add any other text.

Output example:
{
"1": 8.5,
"2": 6,
"3": 3
}
```

G.3 Results by external predictors

As shown in Figure 8(a), PepMind maintains a favorable property profile under all four external predictors, including comparisons with the agentic baselines Biomni and STELLA. It achieves a high AMP-CLIP score and ToxTeller non-toxicity score, while producing low AMP-READ and ConsAMP-Hemo values, indicating stronger predicted potency and lower hemolytic risk, respectively. Figure 8(b) further shows that PepMind achieves the highest overall hypervolume and the best result in 9 of the 10 pairwise and triple objective combinations. The corresponding harmonic means in Figure 8(c) show a similar pattern, with PepMind ranking first in 9 of the 10 combinations. These results indicate that PepMind retains its advantage under independent predictors and remains competitive against both specialized AMP generation methods and general agentic systems.

H Complete Harmonic-Mean

Table 6 presents the complete harmonic-mean comparison across all seven pathogen-specific AMP design tasks. We report scores for six two-objective subsets, AH, AT, AP, HT, HP, and TP, and four

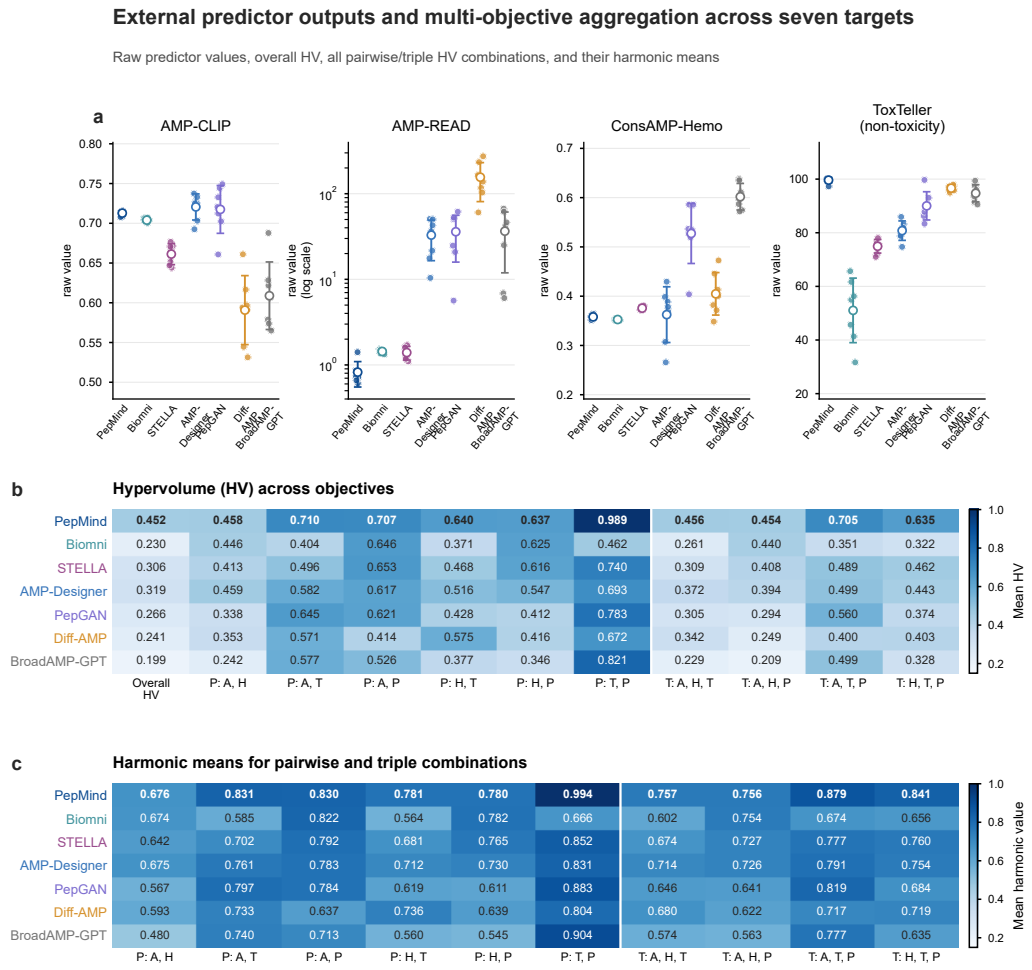

Figure 8: External evaluation using independent AMP property predictors across seven pathogen-specific tasks and seven generation methods, including the agentic baselines Biomni and STELLA. (a) Raw outputs of AMP-CLIP, AMP-READ, ConsAMP-Hemo, and ToxTeller. Each point represents one target task, while the open marker and error bar show the mean and standard deviation across seven tasks. AMP-READ is shown on a logarithmic scale. (b) Mean hypervolume for the full objective set and all pairwise and triple objective combinations. (c) Harmonic means of the corresponding pairwise and triple objective scores. In (b) and (c), the prefixes "P:" and "T:" denote pairwise and triple objective combinations, respectively. All objectives are aligned so that higher values indicate better performance.

three-objective subsets, AHT, AHP, ATP, and HTP. In contrast to the arithmetic mean, the harmonic mean is strongly influenced by the lowest-valued objective. It therefore measures whether a method achieves balanced performance across the objectives in each subset rather than relying on a single dominant property.

For two direction-aligned objective values x and y, the pairwise harmonic-mean score is defined as

$$\mathrm{PairScore}(x, y) = \frac{2xy}{x + y + \epsilon}, \tag{48}$$

where ϵ is a small positive constant introduced for numerical stability. The six two-objective scores are therefore computed as

$$\begin{aligned} \mathrm{AH} &= \mathrm{PairScore}(A, H), \quad \mathrm{AT} = \mathrm{PairScore}(A, T), \\ \mathrm{AP} &= \mathrm{PairScore}(A, P), \quad \mathrm{HT} = \mathrm{PairScore}(H, T), \\ \mathrm{HP} &= \mathrm{PairScore}(H, P), \quad \mathrm{TP} = \mathrm{PairScore}(T, P). \end{aligned} \tag{49}$$

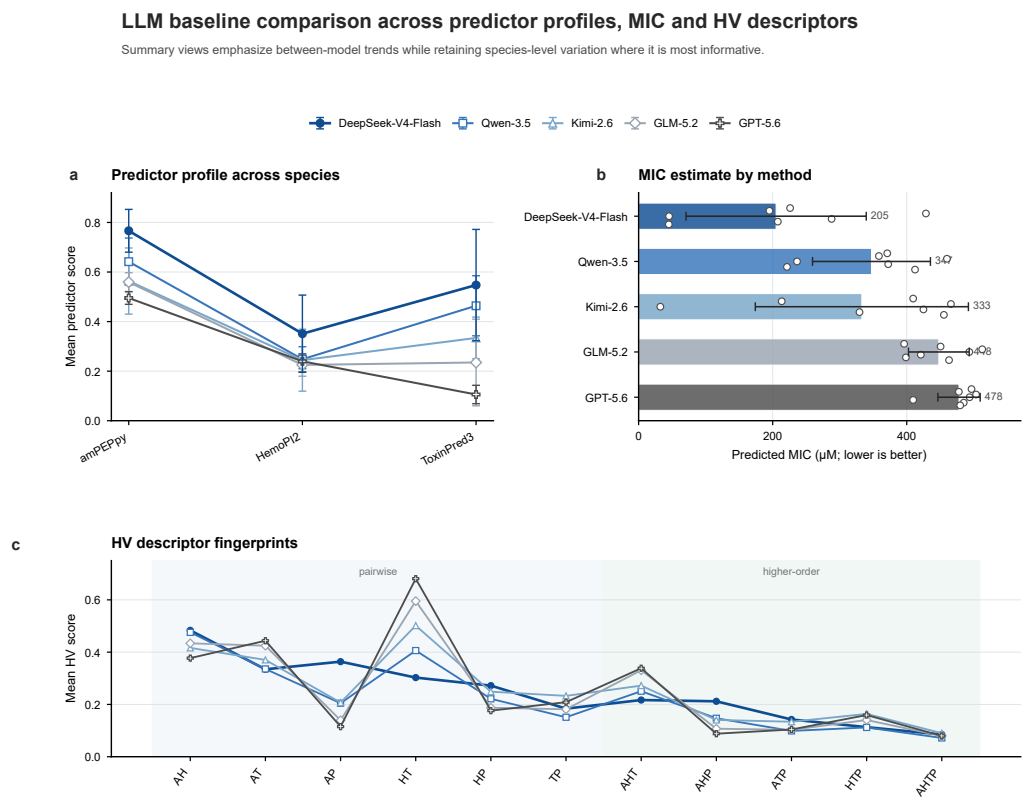

Figure 9: External comparison with general-purpose LLM baselines. (a) Mean activity and safety predictor profiles of peptides generated by five LLMs across target species. (b) Predicted target-specific MIC, where lower values indicate stronger predicted potency. (c) Pairwise and higher-order hypervolume profiles across activity (A), hemolytic safety (H), toxicity safety (T), and target-specific potency (P).

For three direction-aligned objective values x, y, and z, the three-objective harmonic-mean score is defined as

$$\mathrm{TripleScore}(x, y, z) = \frac{3}{\frac{1}{x+\epsilon} + \frac{1}{y+\epsilon} + \frac{1}{z+\epsilon}}, \tag{50}$$

and the corresponding scores are calculated as

$$\begin{aligned} \mathrm{AHT} &= \mathrm{TripleScore}(A, H, T), \\ \mathrm{AHP} &= \mathrm{TripleScore}(A, H, P), \\ \mathrm{ATP} &= \mathrm{TripleScore}(A, T, P), \\ \mathrm{HTP} &= \mathrm{TripleScore}(H, T, P). \end{aligned} \tag{51}$$

As shown in Table 6, PepMind ranks first in 42 of the 70 pathogen–objective-subset comparisons. When the sequential variant is also considered, a PepMind-based method ranks first in 51 cases. PepMind performs especially well on combinations involving efficacy and safety, achieving the best AH and AT scores on all seven pathogens, the best AHT score on six pathogens, and the best HTP score on five pathogens.

Some external baselines remain competitive on potency-related subsets such as AP, TP, and ATP. However, their gains are less stable when hemolytic and toxicity safety are also considered. PepMind performs more consistently on objective combinations that require both efficacy and safety. These results complement the hypervolume analysis. They show that PepMind improves not only Pareto-set coverage, but also the balance among individual objectives.

Table 6: Complete comparison of harmonic-mean scores across seven pathogen-specific AMP design tasks. The best result for each task and objective subset is shown in bold.

Method	AH	AT	AP	HT	HP	TP	AHT	AHP	ATP	HTP
Task 1: *A. baumannii*										
PepMind	**0.8006**	**0.8431**	0.6972	**0.8628**	0.7105	0.7436	**0.8343**	0.7325	0.7755	**0.7661**
PepMind (sequential)	0.7683	0.8145	**0.8086**	0.7449	0.7399	**0.7827**	0.7748	**0.7712**	**0.8017**	0.7553
Biomni	0.6787	0.4659	0.2220	0.5275	0.2351	0.2030	0.5439	0.2933	0.2592	0.2709
STELLA	0.7117	0.6909	0.5924	0.7340	0.6238	0.6078	0.7118	0.6388	0.6275	0.6506
AMP-Designer	0.7795	0.7962	0.7487	0.7943	**0.7460**	0.7607	0.7889	0.7564	0.7659	0.7644
BroadAMP-GPT	0.3956	0.7004	0.7576	0.3864	0.4237	0.7795	0.4468	0.4824	0.7272	0.4679
PepGAN	0.5648	0.7129	0.7748	0.5482	0.5693	0.7389	0.5836	0.6069	0.7280	0.5851
Diff-AMP	0.6044	0.7214	0.2657	0.7688	0.2490	0.2872	0.6826	0.3052	0.3390	0.3252
Task 2: *P. aeruginosa*										
PepMind	**0.8009**	**0.8417**	0.6740	**0.8594**	**0.6852**	0.7135	**0.8329**	**0.7149**	0.7354	**0.7444**
PepMind (sequential)	0.7030	0.7639	0.7264	0.6528	0.6252	0.6730	0.7037	0.6820	0.7192	0.6498
Biomni	0.7079	0.5429	0.4243	0.5954	0.4556	0.3811	0.6080	0.5030	0.4397	0.4617
STELLA	0.7000	0.7061	0.6168	0.7485	0.6489	0.6542	0.7176	0.6535	0.6570	0.6809
AMP-Designer	0.6132	0.7453	**0.7654**	0.6084	0.6097	**0.7487**	0.6359	0.6401	**0.7465**	0.6319
BroadAMP-GPT	0.4019	0.7611	0.6677	0.4288	0.3979	0.7452	0.4868	0.4600	0.7176	0.4821
PepGAN	0.5070	0.6061	0.7500	0.4149	0.5065	0.6070	0.4674	0.5574	0.6181	0.4655
Diff-AMP	0.5553	0.6665	0.2988	0.7525	0.3136	0.3453	0.6410	0.3582	0.3861	0.4027
Task 3: *K. pneumoniae*										
PepMind	**0.8070**	**0.8757**	0.6615	**0.8709**	**0.6584**	**0.7037**	**0.8496**	**0.7020**	**0.7354**	**0.7332**
PepMind (sequential)	0.6699	0.7696	0.6679	0.6098	0.5441	0.6081	0.6769	0.6214	0.6755	0.5857
Biomni	0.6924	0.5243	0.4455	0.5925	0.4939	0.4019	0.5953	0.5251	0.4518	0.4838
STELLA	0.7005	0.6899	0.5271	0.7469	0.5597	0.5529	0.7116	0.5869	0.5819	0.6080
AMP-Designer	0.5448	0.7087	**0.7615**	0.5171	0.5171	0.6804	0.5614	0.5716	0.7029	0.5414
BroadAMP-GPT	0.4459	0.8114	0.6135	0.4866	0.3991	0.6890	0.5394	0.4656	0.6921	0.4934
PepGAN	0.5809	0.7865	0.6677	0.6102	0.5102	0.6850	0.6330	0.5630	0.7016	0.5731
Diff-AMP	0.5480	0.6664	0.4178	0.7448	0.4459	0.5158	0.6346	0.4611	0.5120	0.5398
Task 4: *S. aureus*										
PepMind	**0.8039**	**0.8490**	0.6702	**0.8388**	**0.6636**	0.6934	**0.8298**	**0.7067**	0.7299	**0.7240**
PepMind (sequential)	0.7809	0.8279	0.5501	0.8229	0.5479	0.5707	0.8100	0.6093	0.6279	0.6260
Biomni	0.6943	0.5161	0.4621	0.5801	0.5128	0.4086	0.5880	0.5401	0.4581	0.4901
STELLA	0.7014	0.6935	0.5277	0.7489	0.5592	0.5542	0.7138	0.5872	0.5835	0.6088
AMP-Designer	0.6680	0.7777	**0.7181**	0.6751	0.5958	0.7092	0.6931	0.6402	0.7292	0.6409
BroadAMP-GPT	0.4178	0.7893	0.6764	0.4472	0.4073	**0.7510**	0.5067	0.4721	**0.7325**	0.4955
PepGAN	0.7079	0.8141	0.5856	0.7852	0.5653	0.6363	0.7629	0.6081	0.6611	0.6444
Diff-AMP	0.6130	0.7044	0.4308	0.7900	0.4597	0.5075	0.6905	0.4878	0.5241	0.5524
Task 5: *E. faecalis*										
PepMind	**0.8061**	**0.8255**	**0.6710**	0.7929	**0.6500**	**0.6606**	0.8063	**0.7021**	**0.7104**	**0.6950**
PepMind (sequential)	0.7937	0.7711	0.5241	0.7451	0.5120	0.5024	0.7694	0.5858	0.5775	0.5676
Biomni	0.6878	0.5073	0.4323	0.5729	0.4790	0.3839	0.5802	0.5124	0.4354	0.4660
STELLA	0.7018	0.7013	0.4664	0.7553	0.4897	0.4894	0.7186	0.5346	0.5344	0.5546
AMP-Designer	0.7849	0.8245	0.5257	**0.8263**	0.5266	0.5422	**0.8105**	0.5900	0.6032	0.6040
BroadAMP-GPT	0.5329	0.8096	0.4745	0.6013	0.3927	0.5248	0.6259	0.4582	0.5708	0.4893
PepGAN	0.7268	0.8251	0.4933	0.8117	0.4885	0.5294	0.7829	0.5490	0.5836	0.5789
Diff-AMP	0.6029	0.6980	0.4256	0.7854	0.4564	0.5059	0.6828	0.4824	0.5197	0.5498
Task 6: *E. coli*										
PepMind	**0.7855**	**0.7988**	0.7171	0.7677	0.6918	0.7093	**0.7989**	0.7250	0.7575	0.7199
PepMind (sequential)	0.5689	0.7413	**0.8241**	0.5086	0.5462	0.7033	0.5914	0.6247	0.7530	0.5749
Biomni	0.6924	0.5295	0.4271	0.5941	0.4682	0.3876	0.5981	0.5066	0.4405	0.4688
STELLA	0.7021	0.6839	0.5517	0.7366	0.5855	0.5728	0.7069	0.6067	0.5976	0.6235
AMP-Designer	0.7778	0.7987	0.7295	**0.7984**	**0.7281**	0.7461	0.7904	**0.7425**	0.7545	**0.7538**
BroadAMP-GPT	0.3892	0.7256	0.7308	0.4045	0.4117	**0.7985**	0.4636	0.4704	0.7430	0.4836
PepGAN	0.5220	0.7510	0.7771	0.5279	0.5353	0.7962	0.5706	0.5791	**0.7687**	0.5825
Diff-AMP	0.6203	0.7089	0.2184	0.7941	0.2223	0.2346	0.6964	0.2775	0.2897	0.2952
Task 7: *E. faecium*										
PepMind	**0.7917**	**0.7753**	0.6813	0.7677	0.6721	0.6625	**0.7975**	0.7096	0.7021	0.6961
PepMind (sequential)	0.7851	0.7680	**0.7963**	0.7327	**0.7585**	0.7425	0.7613	**0.7797**	**0.7683**	0.7445
Biomni	0.6866	0.4969	0.4443	0.5613	0.4951	0.3882	0.5713	0.5239	0.4386	0.4704
STELLA	0.6904	0.7003	0.4930	0.7532	0.5186	0.5242	0.7136	0.5550	0.5593	0.5810
AMP-Designer	0.7838	0.7750	0.7467	**0.8074**	0.7579	**0.7704**	0.7943	0.7603	0.7677	**0.7759**
BroadAMP-GPT	0.3914	0.5575	0.7708	0.3410	0.4351	0.6614	0.3971	0.4872	0.6242	0.4277
PepGAN	0.6007	0.7038	0.7865	0.5705	0.6079	0.7377	0.5998	0.6407	0.7204	0.6043
Diff-AMP	0.5476	0.7502	0.3923	0.6223	0.3131	0.4317	0.6166	0.3748	0.4700	0.3938

I Detailed Memory Ablation Results

Table 7 reports the complete memory ablation results across seven pathogen-specific tasks and 11 objective subsets. All variants use the same optimization pipeline, candidate budget, predictors, and evaluation protocol. They differ only in how historical records are used. *Full* retrieves relevant

Table 7: Memory ablation results across seven pathogen-specific AMP design tasks. The best result for each objective subset within each task is shown in bold.

Memory Condition	AH	AT	AP	HT	HP	TP	AHT	AHP	ATP	HTP	AHTP
					E. faecalis						
Full	**0.9844**	**0.9559**	**0.6695**	0.9611	**0.6721**	**0.6442**	**0.9459**	**0.6602**	**0.6303**	**0.6341**	**0.6231**
None	0.8883	0.6536	0.5204	0.2878	0.3091	0.2964	0.2641	0.2879	0.2717	0.0941	0.0864
Success-only	0.9624	0.8315	0.6418	0.8457	0.6458	0.5593	0.8238	0.6294	0.5445	0.5475	0.5342
Failure-only	0.9710	0.9446	0.6139	**0.9666**	0.5799	0.6027	0.9316	0.5573	0.5790	0.5414	0.5200
Random	0.9521	0.8927	0.6221	0.9029	0.6262	0.5745	0.8754	0.6057	0.5551	0.5594	0.5410
					K. pneumoniae						
Full	0.8882	**0.9307**	0.9077	0.9021	**0.9146**	**0.9548**	0.8552	**0.8320**	**0.8674**	**0.8683**	**0.7918**
None	0.8447	0.7022	0.7663	0.6099	0.6364	0.6881	0.4735	0.4883	0.4752	0.2864	0.2108
Success-only	0.9175	0.9301	0.7312	0.9570	0.7476	0.7471	**0.9042**	0.7000	0.7057	0.7214	0.6774
Failure-only	**0.9234**	0.9238	**0.9443**	**0.9697**	0.7750	0.8021	0.8738	0.7203	0.7568	0.6822	0.6175
Random	0.8824	0.8521	0.7121	0.9356	0.7431	0.7425	0.8336	0.6654	0.6640	0.6995	0.6235
					S. aureus						
Full	0.8844	0.8815	0.8164	0.9918	**0.8855**	**0.9092**	0.8714	**0.7751**	**0.7894**	**0.8651**	**0.7507**
None	0.7793	0.7757	**0.8407**	0.4240	0.7817	0.6503	0.3293	0.6347	0.5760	0.3206	0.2494
Success-only	0.9438	**0.9548**	0.7583	0.9900	0.7820	0.7658	**0.9352**	0.7388	0.7301	0.7577	0.7147
Failure-only	**0.9697**	0.7637	0.6843	0.7874	0.6912	0.4351	0.7488	0.6498	0.4131	0.4225	0.3996
Random	0.9306	0.9323	0.7060	**0.9937**	0.7400	0.7420	0.9262	0.6881	0.6885	0.7223	0.6703
					A. baumannii						
Full	0.8927	0.8914	0.7235	**0.9985**	0.8097	0.8094	**0.8889**	0.7196	0.7161	**0.8100**	**0.7115**
None	0.8058	0.7376	0.9180	0.7274	0.7517	0.5496	0.6004	0.6273	0.4472	0.4334	0.3669
Success-only	0.8921	0.8662	0.7156	0.9710	0.8078	0.7931	0.8625	0.7112	0.6902	0.7888	0.6877
Failure-only	**0.9087**	**0.9226**	**0.9809**	0.9123	**0.8508**	**0.9337**	0.8593	**0.8176**	**0.8877**	0.7532	0.7092
Random	0.8332	0.7483	0.5184	0.9041	0.6247	0.5608	0.7394	0.5148	0.4565	0.5554	0.4517
					E. coli						
Full	**0.9897**	**0.9221**	0.7854	0.9428	0.8025	0.7414	**0.9078**	0.7725	0.7052	0.7044	**0.6672**
None	0.7824	0.7933	0.7678	0.8554	0.7841	**0.9193**	0.5877	0.6050	0.6948	0.5731	0.4140
Success-only	0.9496	0.8881	0.7645	0.9284	0.7821	0.7162	0.8754	0.7410	0.6806	0.6963	0.6570
Failure-only	0.9592	0.8863	**0.9450**	**0.9518**	**0.9302**	0.7793	0.8317	**0.8797**	**0.7399**	**0.7247**	0.6660
Random	0.9512	0.8878	0.7346	0.9213	0.7292	0.6576	0.8728	0.6914	0.6197	0.6286	0.5892
					E. faecium						
Full	0.8813	0.8307	0.7490	0.9610	0.8246	0.7329	0.8226	**0.7333**	0.6398	**0.7186**	**0.6240**
None	**0.9393**	0.7196	**0.9272**	0.4511	0.2863	**0.9014**	0.1627	0.2686	**0.7153**	0.1263	0.0947
Success-only	0.8837	**0.8626**	0.6872	0.9732	0.7797	0.6948	**0.8587**	0.6792	0.6031	0.6883	0.5990
Failure-only	0.7869	0.7409	0.6124	**0.9856**	**0.8309**	0.6132	0.7349	0.6091	0.4494	0.6113	0.4433
Random	0.8860	0.8223	0.6362	0.9224	0.7009	0.6311	0.8132	0.6216	0.5586	0.6144	0.5438
					P. aeruginosa						
Full	0.9072	0.8983	0.7625	0.9887	0.8488	0.8384	**0.8989**	0.7565	**0.7465**	0.8306	**0.7392**
None	0.7844	0.4650	0.8302	0.3379	0.8877	0.3947	0.2481	0.6184	0.2867	0.2544	0.1699
Success-only	0.9064	0.8995	0.7580	0.9873	0.8333	0.8164	0.8974	0.7537	0.7403	0.8121	0.7361
Failure-only	**0.9614**	**0.9249**	**0.9323**	0.9835	**0.9300**	0.7451	0.8369	**0.8918**	0.7023	0.6612	0.5665
Random	0.8156	0.8147	0.7533	**0.9951**	0.9134	**0.8877**	0.8096	0.7446	0.7223	**0.8764**	0.7126

successful and failed records. *None* removes long-term memory. *Success-only* and *Failure-only* use one record type, while *Random* retrieves memory records without relevance ranking. In this ablation study, the memory pool contains experience from all seven pathogens. This broader memory pool may partly explain the stronger results than those reported in the main-text setting.

All memory-enabled variants outperform the no-memory setting under the complete AHTP objective, with mean HV ranging from 0.5603 to 0.7011, compared with 0.2274 for no memory. Full memory achieves the highest mean AHTP HV (0.7011), followed by success-only memory (0.6580), both exceeding random memory (0.5903). Across all objective subsets, full memory ranks first in 37 of 77 comparisons, including 24 of 35 three- and four-objective comparisons, and achieves the best AHTP result on all seven pathogens. Success-only and failure-only memory remain competitive on some subsets, suggesting complementary roles: successful records provide effective design patterns, while failed records help identify bottlenecks and avoid repeated mistakes. Together, these results support the value of long-term memory and relevant retrieval.

J Memory Baselines

Comparison with Memory Baselines. Figure 10 further decomposes the HV gains of full memory over the no-memory and random-memory baselines across individual objective subsets. Compared with no memory, the largest improvements are observed on AHT ($+0.504$), HTP ($+0.478$), AHTP ($+0.474$), and HT ($+0.436$), showing that memory becomes increasingly important when several efficacy and safety objectives must be optimized jointly. Full memory also outperforms random

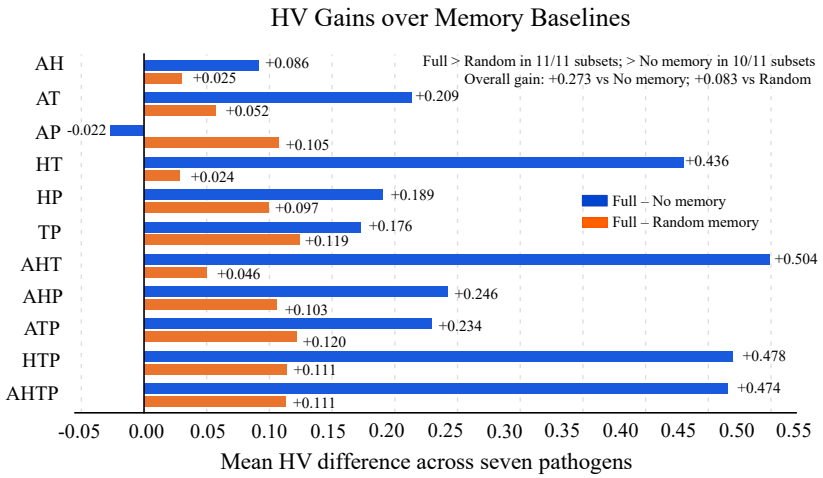

Figure 10: HV gains of full memory over the two memory baselines across all objective subsets.

memory on every objective subset, with an overall mean gain of 0.083. This consistent advantage indicates that the benefit of memory does not arise simply from retaining previous examples. Instead, retrieving records that are relevant to the current optimization state provides more effective guidance for multi-objective peptide design.

K Prompt Details

PepMind uses a multi-stage prompting workflow consisting of task parsing, experience retrieval, strategy planning, candidate generation, evaluation, and iterative refinement. To keep the appendix concise, we report only the three prompt components most directly involved in peptide design and refinement.

K.1 External LLM Evaluation Prompt

```
System:

You are an expert in antimicrobial peptide (AMP) drug design. Please
    evaluate the overall potential of each peptide sequence as an
    antimicrobial drug candidate.

User:
1. Antimicrobial activity: content of positively charged residues (K/R),
    hydrophobicity pattern, amphipathicity, and known AMP motifs.

2. Safety: risk of hemolysis (excessive hydrophobicity indicates higher
    risk) and cytotoxicity.

3. Structural plausibility: sequence length (8--35 amino acids is
    considered optimal) and amino-acid composition.

4. Developability: the overall ability to balance antimicrobial activity
    and safety.

Assign each sequence a score from 1 to 10, where 10 indicates the most
    promising AMP candidate.
Strictly output a single JSON object in which each key is the string-form
     input index and each value is the corresponding score. The output
    must contain all input indices and must not include any additional
    text.
```

```
Example:
{"1":8.5,"2":6,"3":3}

Score interpretation:


A (>=8): Excellent; high development potential.
B (5--7): Good; has room for optimization.
C (<=4): Poor; not suitable as an AMP candidate.
Evaluation criteria:
```

K.2 Strategy Planning Prompt

```
System:
You are an expert in antimicrobial peptide design.
Construct a balanced design strategy from the task,
constraints, and retrieved historical experience.
Return valid JSON only.

User:
Target pathogen: <TARGET_PATHOGEN>
Length range: <MIN_LENGTH>-<MAX_LENGTH> aa
Number of candidates: <CANDIDATE_NUMBER>
APEX MIC threshold: <= <MIC_THRESHOLD> uM

Successful experiences:
<SUCCESSFUL_EXPERIENCES>

Failure-associated experiences:
<FAILURE_EXPERIENCES>

Statistical patterns derived from retrieved experiences:

Successful-pattern statistics:
<SUCCESS_PATTERN_STATISTICS>

Failure-pattern statistics:
<FAILURE_PATTERN_STATISTICS>

Recommendations derived from their comparison:
<CONTRASTIVE_RECOMMENDATIONS>

Use successful patterns as positive design guidance and
failure-associated patterns as explicit avoidance guidance.
Integrate both types of evidence into one balanced strategy.

Provide at least ten design principles covering activity,
hemolysis, toxicity, target-specific MIC, charge,
hydrophobicity, amphipathicity, sequence length, and
sequence diversity. All four objectives must be considered
jointly, and repeated failure patterns must be explicitly
addressed.

Output:
{
"strategy": {
"goal": "<DESIGN_GOAL>",
"generation_principles": [
"<PRINCIPLE_1>",
"<PRINCIPLE_2>",
"<ADDITIONAL_PRINCIPLES>"
]
}
```

```
}
```

K.3 CANDIDATE-GENERATION PROMPT

```
System:
You are an expert in antimicrobial peptide sequence design.

Generate candidate peptide sequences by following the
provided design strategy and evaluation feedback.

Return valid JSON only. Do not include Markdown code fences
or any text outside the JSON object. Use only the 20
standard amino acids:
ACDEFGHIKLMNPQRSTVWY.

Do not generate duplicate sequences.

User:
Previous-round feedback:
<ROUND_SPECIFIC_FEEDBACK>

Current task:
Target pathogen: <TARGET_PATHOGEN>
Length range: <MIN_LENGTH>-<MAX_LENGTH> aa
Number of candidates: <CANDIDATE_NUMBER>

The sequence-length range is a hard constraint. Return
exactly <CANDIDATE_NUMBER> valid and non-duplicate
sequences.

A candidate is successful only when all four constraints
are satisfied simultaneously:

amPEPpy >= 0.70
HemoPI2 <= 0.40
ToxinPred3 <= 0.40
APEX_MIC_target <= <MIC_THRESHOLD> uM

Design strategy:
<GENERATION_PRINCIPLES>

Use the provided strategy as the primary guidance for
candidate generation. Do not reformulate or replace the
strategy.

Retrieved successful experiences:
<SUCCESSFUL_EXPERIENCES>

Use these records as positive sequence-level evidence.
Preserve transferable patterns that are consistent with
the provided strategy, but do not directly copy complete
historical sequences.

Retrieved failure-associated experiences:
<FAILURE_EXPERIENCES>

Use these records as explicit avoidance evidence. Avoid
repeating sequence motifs or physicochemical patterns
associated with activity, hemolysis, toxicity, or MIC
failures.

Generate candidates with diverse residue arrangements and
physicochemical profiles. Use feedback from the preceding
```

```
round to make targeted modifications while preserving
objectives that already satisfy their thresholds. Do not
improve one objective by sacrificing another.

For each candidate, provide a concise design rationale,
identify the historical records that substantively
influenced its design, and assess its potential risks
across all four objectives.

Output:
{
  "candidates": [
    {
      "id": "cand_001",
      "sequence": "<SEQUENCE>",
      "rationale": "<DESIGN_RATIONALE>",
      "evidence_ids": ["<MEMORY_ID>"],
      "risk_note": "<ACTIVITY, HEMOLYSIS, TOXICITY, AND MIC
                    RISK ANALYSIS>"
    }
  ]
}
```

K.4 Iterative Feedback Template

```
Round <ROUND_ID>:

Accepted: <ACCEPTED>/<TOTAL>
Activity failures: <ACTIVITY_FAILURES>
Hemolysis failures: <HEMOLYSIS_FAILURES>
Toxicity failures: <TOXICITY_FAILURES>
MIC failures: <MIC_FAILURES>
Diversity: <DIVERSITY_SCORE>

Analyze the failed candidates with respect to sequence
length, K/R density, hydrophobic-residue fraction,
hydrophobic clusters, repeated motifs, and net charge.

Activity-related patterns:
<ACTIVITY_FAILURE_PATTERNS>

Hemolysis-related patterns:
<HEMOLYSIS_FAILURE_PATTERNS>

Toxicity-related patterns:
<TOXICITY_FAILURE_PATTERNS>

MIC-related patterns:
<MIC_FAILURE_PATTERNS>

Best current candidate:
<BEST_CURRENT_CANDIDATE>

Best historical candidate:
<BEST_HISTORICAL_CANDIDATE>

Produce targeted modifications that improve the failed
objectives without degrading objectives that already meet
their thresholds. Use successful candidates as local
templates and modify a small number of residue positions.
```