LaPep: Can Language Contribute to Property-Guided Peptide Design?

Kimberly Liang,[1,*] **Tong Chen,**[1] **Pranam Chatterjee**[1,2,†]

[1]Department of Computer and Information Science, University of Pennsylvania
[2]Department of Bioengineering, University of Pennsylvania

[†]Corresponding author: pranam@seas.upenn.edu

Abstract

Large language models (LLMs) encode broad chemical heuristics from the scientific literature and are increasingly proposed as tools for therapeutic molecule design. However, their effectiveness for generating therapeutically viable peptides, particularly in the absence of strong labeled predictors, remains unclear. We introduce **LaPep**, a sampling-time framework that integrates LLMs as token-level proposers within a discrete flow-based peptide generator, while using hard property predictors to guide and evaluate generation. Using open-source LLMs including Qwen3, Kimi K2, and Llama 3, we study two representative design settings: permeability, where a strong predictor exists, and protease stability, where it does not. We show that language guidance can improve permeability when combined with a hard predictor, but provides limited or inconsistent gains for protease stability when used alone, despite leveraging external heuristic scorers. These results highlight that current LLMs are not yet reliable substitutes for quantitative property models in therapeutic peptide design. We position LaPep as a strong diagnostic framework for systematically evaluating the capabilities and limitations of language models in guided molecular generation, and argue that high-quality labeled predictors remain critical for translating language-driven design into therapeutically relevant outcomes.

1 Introduction

Discrete generative models have emerged as a powerful paradigm for molecular design (Lee et al., 2025), particularly when coupled with explicit property guidance. For example, recent frameworks based on masked discrete diffusion (Tang et al., 2025a;c; Vincoff et al., 2025) and discrete flow matching (Chen et al., 2025b;a) enable peptide sequences to be steered toward desired objectives using trained property predictors, supporting multi-objective optimization over binding, toxicity, permeability, and related therapeutic criteria (Zhang et al., 2026). When reliable labeled data is available, these approaches provide precise, interpretable control over generation and have demonstrated strong empirical performance both *in silico* (Tang et al., 2025a;c; Chen et al., 2025a) and *in vitro* (Chen et al.). However, their effectiveness fundamentally depends on the existence of accurate predictors (Zhang et al., 2026; Wang et al., 2016; Swanson et al., 2024), which are often unavailable for more nuanced or context-dependent properties such as protease stability, formulation robustness, or degradation pathways.

Natural language offers an appealing alternative source of design information, as large language models encode broad chemical and medicinal heuristics from the scientific literature (Bran et al., 2023; Narayanan et al., 2025; Jablonka et al., 2024). In practice, language is frequently used by practitioners to reason about missing objectives during early-stage therapeutic development, long before sufficient data exists to train predictors (Zheng et al., 2024). This makes peptide generation under incomplete supervision a natural and realistic setting in which language models might be expected to contribute. At the same time, language-derived signals are inherently uncalibrated, context dependent, and difficult to reconcile with hard feasibility constraints Shrivastava et al. (2025). This raises a central question: *can current natural language models meaningfully assist guided peptide generation, and when do quantitative property models remain indispensable?*

In this work, we introduce **LaPep**, a controlled framework that integrates language-driven proposals with predictor-constrained discrete peptide generation, enabling systematic evaluation of language guidance in settings with and without strong property supervision. By studying permeability as a property with a reliable predictor and protease stability as one without, LaPep mirrors how language is often used in practice and provides a diagnostic lens on the current capabilities and limitations of language models for therapeutic peptide design. The results highlight the continued importance of high-quality labeled data and motivate closer collaboration between generative modeling and experimental measurement.

Our contributions are threefold:

1. **A controlled framework for evaluating language-guided peptide design.** We introduce LaPep as a modular system that combines language-driven proposal mechanisms with predictor-constrained discrete generative models, isolating the effect of language guidance under realistic therapeutic design conditions.
2. **An empirical analysis of language guidance under incomplete supervision.** We study language-assisted peptide generation in settings with strong property predictors and in settings where predictors are unavailable, providing a clear comparison of when language helps and when it does not.
3. **A diagnostic perspective on the role of data in therapeutic design.** The results demonstrate that language models are not yet substitutes for quantitative property predictors and underscore the need for labeled experimental data to enable reliable, guided peptide generation.

2 Related Works

Property-Guided Discrete Molecule Design Discrete diffusion and discrete flow matching have emerged as effective non-autoregressive frameworks for molecular generation in sequence space (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024; Gat et al., 2024; Tang et al., 2025b; Peng et al., 2025). In the small-molecule setting, variants such as GenMol (Lee et al., 2025) can be extended to incorporate classifier- or constraint-based guidance to enforce feasibility during discrete diffusion (Cardei et al.). For peptides, recent methods including PepTune, TR2-D2 MOG-DFM, and AReUReDi demonstrate that discrete diffusion and flow matching can be steered toward multiple therapeutic objectives using trained property predictors (Tang et al., 2025a;c; Chen et al., 2025b;a; Goel et al., 2025; Rector-Brooks et al., 2025). As extensions of MOG-DFM and PepTune, models such as moPPIt and SOAPIA enable highly-specific peptide design with specialized predictors on carefully-curated molecular interaction data (Chen et al.; Vincoff et al., 2025). These approaches provide precise control when reliable predictors are available, but fundamentally rely on labeled supervision for each target property, motivating investigation of complementary guidance signals when such predictors are missing or unreliable.

Natural Language-Guided Molecular and Protein Design. Recent work has explored using natural language as a conditioning signal for molecular and protein generation. In the protein domain, methods such as Pinal, BioM3, and ProteinDT align protein sequences with textual descriptions through joint embedding or contrastive objectives, enabling language-conditioned sequence generation without explicit property constraints (Dai et al., 2024; Praljak et al., 2024; Liu et al., 2025). InstructPro extends this paradigm to ligand-binding protein design by conditioning generation on natural language descriptions of binding behavior (Song et al., 2025). While these approaches demonstrate that language can influence generated sequences, they typically rely on language as a direct conditioning or scoring signal and do not incorporate hard feasibility constraints or quantitative property predictors during generation. As a result, the extent to which language guidance produces therapeutically viable proteins or peptides, particularly in the absence of labeled property supervision, remains an open question.

3 Methods

3.1 Problem Setup

Let $\mathcal{V}$ be a finite vocabulary and let $x = (x_1, \ldots, x_L) \in \mathcal{V}^L$ denote a discrete sequence of length L. We consider N objectives $s_n : \mathcal{V}^L \to \mathbb{R}$ and write $s(x) = (s_1(x), \ldots, s_N(x))^\top$. Some

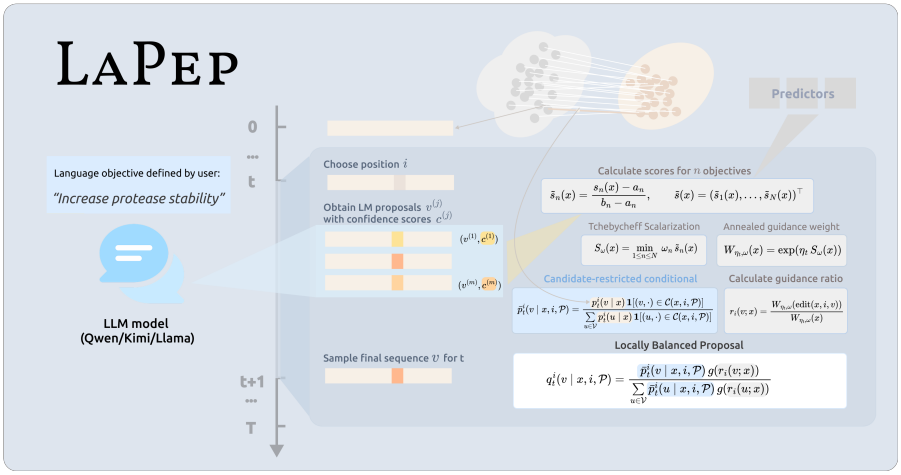

Figure 1: **Overview of LaPep.** A language model proposes a restricted set of single-site substitutions with confidence scores for a user-specified natural-language objective, and these scores are incorporated as objectives within the AReUReDi-style multi-objective guided discrete flow matching sampler to update the peptide sequence over time.

objectives admit direct predictors, while others are specified in natural language and lack reliable hard evaluators. We assume access to a discrete flow matching base generator that defines a sequence prior $p_0(x)$ and provides positionwise token conditionals used to propose single-site edits. The goal is to sample sequences that achieve strong multi-objective tradeoffs while incorporating language-specified desiderata as additional objective coordinates. Our construction follows the same four-part organization and the same locally balanced sampling principle as AReUReDi (Chen et al., 2025a), with the only modification that language-model confidence scores are treated as objectives and language-model suggestions restrict the set of candidate edits evaluated at each step (Figure 1).

3.2 Annealed multi objective guidance

We map heterogeneous objectives to a comparable scale using an affine normalization that targets the unit interval. For each objective index n, let a_n and b_n be running lower and upper calibration statistics computed from visited samples with $b_n > a_n$, and define

$$\tilde{s}_n(x) = \frac{s_n(x) - a_n}{b_n - a_n}, \qquad \tilde{s}(x) = (\tilde{s}_1(x), \dots, \tilde{s}_N(x))^\top. \tag{1}$$

Given weights $\omega \in \Delta^{N-1}$, we use the Tchebycheff scalarization adopted in AReUReDi,

$$S_\omega(x) = \min_{1 \le n \le N} \omega_n \, \tilde{s}_n(x). \tag{2}$$

We convert this scalarization into an annealed guidance weight using a strength parameter $\eta_t > 0$ at iteration t,

$$W_{\eta_t,\omega}(x) = \exp(\eta_t \, S_\omega(x)), \tag{3}$$

and define the guided target distribution

$$\pi_t(x) \propto p_0(x) \, W_{\eta_t,\omega}(x). \tag{4}$$

Increasing η_t over iterations sharpens concentration toward sequences with improved worst-case weighted objective performance under S_ω while allowing exploration at earlier iterations.

3.3 Language objectives and locally balanced candidate restricted proposals

LaPep uses a language model $\mathcal{M}$ to operationalize objectives that lack hard predictors. Let $\mathcal{P}$ denote an engineered textual prompt and let $i \in \{1, \dots, L\}$ be the position selected for mutation. Conditioned on x, i, and $\mathcal{P}$, the language model returns a finite set of replacement tokens and confidence scores

$$\mathcal{C}(x, i, P) = \{(v^{(1)}, c^{(1)}), \dots, (v^{(m)}, c^{(m)})\}, \qquad v^{(j)} \in \mathcal{V}, \;\; c^{(j)} \in [0, 1]. \tag{5}$$

Table 1: **Language guidance does not add benefit when a hard predictor is available.** We compare mean PAMPA and Caco2 permeability scores under hard-predictor-only, language-only, combined hard predictor plus language, and unconditional generation, all evaluated by the external CPMP model, with lower being better and best highlighted in bold.

Settings	PAMPA (↓)	Caco2 (↓)
Hard Predictor Only	**-7.01**	-6.28
Language Model only	-6.81	-6.1
Hard Predictor + Language Model	-6.58	**-6.71**
Unconditional	-6.8	-6.38

Each token $v^{(j)}$ induces a single-site edit $y^{(j)} = \mathrm{edit}(x, i, v^{(j)})$. We interpret each $c^{(j)}$ as a quantitative score for a language-defined criterion associated with $\mathcal{P}$ and include it as one or more coordinates of the objective vector $s(x)$. This inclusion only changes the evaluation of $\tilde{s}(x)$ and thus the value of $W_{\eta_t,\omega}(x)$ through Eqs. equation 1 - equation 3.

At iteration t, let $p_t^i(v \mid x)$ denote the base generator conditional probability of placing token v at position i given the current sequence x. LaPep restricts candidate evaluation by using the language-model candidate set as a mask. Define the candidate-restricted conditional

$$\bar{p}_t^i(v \mid x, i, \mathcal{P}) = \frac{p_t^i(v \mid x)\,\mathbf{1}[(v, \cdot) \in \mathcal{C}(x, i, \mathcal{P})]}{\sum_{u \in \mathcal{V}} p_t^i(u \mid x)\,\mathbf{1}[(u, \cdot) \in \mathcal{C}(x, i, \mathcal{P})]}. \tag{6}$$

For any token v in the support of $\bar{p}_t^i(\cdot \mid x, i, \mathcal{P})$, define the guided ratio for the corresponding edited state $y = \mathrm{edit}(x, i, v)$ as

$$r_i(v; x) = \frac{W_{\eta_t,\omega}(\mathrm{edit}(x, i, v))}{W_{\eta_t,\omega}(x)}. \tag{7}$$

Let $g : \mathbb{R}_+ \to \mathbb{R}_+$ be a balancing function satisfying $g(u) = u\, g(1/u)$. The locally balanced proposal over replacement tokens at position i is

$$q_t^i(v \mid x, i, \mathcal{P}) = \frac{\bar{p}_t^i(v \mid x, i, \mathcal{P})\, g(r_i(v; x))}{\sum_{u \in \mathcal{V}} \bar{p}_t^i(u \mid x, i, \mathcal{P})\, g(r_i(u; x))}. \tag{8}$$

This proposal has the same locally balanced form as in AReUReDi (Chen et al., 2025a) while restricting computation to candidates suggested by the language model.

3.4 Sampling update and prompt specification

Given the current sequence x_t, we select a position index i from a scan distribution over positions and sample a replacement token v from $q_t^i(\cdot \mid x_t, i, \mathcal{P})$. We set $x_{t+1} = \mathrm{edit}(x_t, i, v)$. Under the locally balanced construction in Eq. equation 8, the single-site update is used in the same acceptance-corrected sampling framework as AReUReDi, and the resulting step can be viewed as an acceptance-one update for the coordinate move when the proposal and target are paired through the balancing function condition (Chen et al., 2025a). In LaPep, the only additional requirement is that the language model must output a parsable candidate set $\mathcal{C}(x, i, \mathcal{P})$ with tokens in the allowed vocabulary and associated confidence scores in the unit interval.

The language model interface is implemented through a fixed prompt template and deterministic parsing rules (Figure S1). Across experiments, the prompt template and parsing are held constant and we vary only the language model instance $\mathcal{M}$. We report results for Qwen3 (Yang et al., 2025), Kimi K2 (Team et al., 2025), and Llama 3 (Grattafiori et al., 2024) by running the full LaPep procedure with one fixed choice of $\mathcal{M}$ per run.

4 Results

We evaluate whether prompted language-model guidance improves peptide property optimization in LaPep under a controlled computational benchmarking pipeline. Across all settings, language

Table 2: **Language-only guidance does not improve protease stability without a hard predictor.** We compare mean protease stability scores for Qwen3-guided language-only optimization versus unconditional generation, evaluated by the external ProsperousPlus platform, with lower being better.

Guidance	Protease Stability (↓)
Language Model	0.5027
Unconditional	0.4225

guidance does not provide reliable gains, either when strong hard predictors are available or when they are not, and this conclusion is robust to language model choice and scale (Tables 1–S3).

We first study the regime where a reliable hard predictor is available and ask whether adding language guidance provides additional benefit. Permeability optimization is evaluated using PAMPA and Caco2 scores computed by the external CPMP model (Jiang et al., 2025). Hard-predictor-only guidance achieves the best PAMPA score, while combining hard predictors with language guidance yields mixed results, improving Caco2 relative to language-only guidance but degrading PAMPA relative to both hard-predictor-only guidance and unconditional generation (Table 1). Language-only guidance does not outperform the strongest non-language baseline and does not consistently improve over unconditional generation. Overall, these results indicate that when a strong predictor is present, language guidance does not provide systematic additive value under external evaluation.

We next consider a setting without access to a pre-trained hard predictor and evaluate whether language guidance alone can improve a property lacking a reliable internal evaluator. Protease stability is assessed using the ProsperousPlus platform (Li et al., 2023). Under this evaluation, Qwen3-guided language-only optimization produces worse stability than unconditional generation, indicating that language guidance fails to improve protease stability (Table 2). This finding shows that the limitation of language guidance is not confined to the predictor-available regime.

We then examine solubility and half-life, where no external evaluation platform is available and metrics are computed by internal predictors. Hard-predictor-only guidance yields the best solubility and predicted half-life, establishing the strongest baseline under this scoring protocol (Table S1). Language-only guidance reduces solubility relative to both hard-predictor-only guidance and unconditional generation, and while it improves predicted half-life relative to unconditional generation, it remains substantially below the hard-predictor baseline, indicating that language-only guidance is not competitive even under internal scoring.

Finally, varying the language model family or scale does not change these conclusions. Solubility under language-only guidance remains below hard-predictor guidance and does not show consistent gains over unconditional generation across Qwen3 (Yang et al., 2025), Llama 3 (Grattafiori et al., 2024), and Kimi K2 (Team et al., 2025) (Table S2). Predicted half-life is non-monotonic across Qwen3 model sizes and remains well below hard-predictor-only guidance, indicating that scaling does not reliably improve alignment with the target property (Table S3).

5 Discussion

Overall, **LaPep** highlights both the promise and the current limitations of using natural language models in therapeutic design. Under the current computational benchmark, prompted language-model guidance via candidate substitutions and confidence scoring does not reliably improve optimization outcomes, either when strong property predictors are available or when they are absent. In settings with reliable predictors, language provides minimal additive benefit beyond hard-predictor guidance, while for objectives lacking labeled supervision, language-only guidance fails to produce consistent gains. These results suggest that, at present, language models are *not reliable substitutes* for quantitative property models in therapeutically relevant peptide design.

Several limitations should be noted. External evaluations rely on proxy platforms such as CPMP for permeability and ProsperousPlus for protease stability, while solubility and half-life are assessed using internal predictors, and agreement between predictors does not guarantee experimental success. Most importantly, definitive assessment of language-guided peptide design ultimately requires *in vivo* validation, as key properties are context dependent and may be dominated by physiological factors

not captured by sequence-level predictors or proxy models. Taken together, our findings emphasize the continued importance of high-quality labeled data and close collaboration with experimentalists to train robust predictors, and provide a realistic framework for evaluating when language guidance can meaningfully complement them.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

Michael Cardei, Jacob K Christopher, Bhavya Kailkhura, Thomas Hartvigsen, and Ferdinando Fioretto. Constrained molecular generation with discrete diffusion for drug discovery. In *NeurIPS 2025 Workshop on AI Virtual Cells and Instruments: A New Era in Drug Discovery and Development*.

Tong Chen, Zachary Quinn, Yinuo Zhang, and Pranam Chatterjee. moppit-v3: Motif-specific peptides generated via multi-objective-guided discrete flow matching. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*.

Tong Chen, Yinuo Zhang, and Pranam Chatterjee. Areuredi: Annealed rectified updates for refining discrete flows with multi-objective guidance. *arXiv preprint arXiv:2510.00352*, 2025a.

Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow matching for controllable biological sequence design. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025b. URL https://openreview.net/forum?id=8YIMLoHP9J.

Fengyuan Dai, Shiyang You, Yudian Zhu, Yuan Gao, Lihao Fu, Xibin Zhou, Jin Su, Chentong Wang, Yuliang Fan, Xiaoxiao Ma, et al. Toward de novo protein design from natural language. *BioRxiv*, pp. 2024–08, 2024.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Shrey Goel, Peregrine Michael Schray, Yinuo Zhang, Sophia Vincoff, Huong T. Kratochvil, and Pranam Chatterjee. Token-level guided discrete diffusion for membrane protein design. In *NeurIPS AI4Science Workshop*, 2025. URL https://openreview.net/forum?id=I0hzddNny7.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169, 2024.

Dawei Jiang, Zixi Chen, and Hongli Du. Cyclic peptide membrane permeability prediction using deep learning model based on molecular attention transformer. *Frontiers in Bioinformatics*, 5: 1566174, 2025.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Yuxing Peng, Saee Gopal Paliwal, Weili Nie, and Arash Vahdat. Genmol: A drug discovery generalist with discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KM7pXWG1xj.

Fuyi Li, Cong Wang, Xudong Guo, Tatsuya Akutsu, Geoffrey I Webb, Lachlan JM Coin, Lukasz Kurgan, and Jiangning Song. Prosperousplus: a one-stop and comprehensive platform for accurate protease-specific substrate cleavage prediction and machine-learning model construction. *Briefings in Bioinformatics*, 24(6):bbad372, 2023.

Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, et al. A text-guided protein design framework. *Nature Machine Intelligence*, pp. 1–12, 2025.

Siddharth M Narayanan, James D Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G Rodriques, and Andrew D White. Training a scientific reasoning model for chemistry. *arXiv preprint arXiv:2506.17238*, 2025.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion models with applications to biological sequence generation. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025. URL https://openreview.net/forum?id=fFuVPKpSt0.

Nikša Praljak, Hugh Yeh, Miranda Moore, Michael Socolich, Rama Ranganathan, and Andrew L Ferguson. Natural language prompts guide the design of novel functional protein sequences. *bioRxiv*, pp. 2024–11, 2024.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Cheng-Hao Liu, Sarthak Mittal, Nouha Dziri, Michael M. Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ombm8S40zN.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Vaishnavi Shrivastava, Ananya Kumar, and Percy Liang. Language models prefer what they know: Relative confidence estimation via confidence preferences. *arXiv preprint arXiv:2502.01126*, 2025.

Zhenqiao Song, Ramith Hettiarachchi, Chuan Li, Jianwen Xie, and Lei Li. Natural language guided ligand-binding protein design. *arXiv preprint arXiv:2506.09332*, 2025.

Kyle Swanson, Parker Walther, Jeremy Leitz, Souhrid Mukherjee, Joseph C Wu, Rabindra V Shivnaraine, and James Zou. Admet-ai: a machine learning admet platform for evaluation of large-scale chemical libraries. *Bioinformatics*, 40(7):btae416, 2024.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=FQoy1Y1Hd8.

Sophia Tang, Yinuo Zhang, Alexander Tong, and Pranam Chatterjee. Gumbel-softmax score and flow matching for discrete biological sequence generation. In *ICLR 2025 Workshop on AI for Nucleic Acids*, 2025b. URL https://openreview.net/forum?id=ITpCmDhSfu.

Sophia Tang, Yuchen Zhu, Molei Tao, and Pranam Chatterjee. Tr2-d2: Tree search guided trajectory-aware fine-tuning for discrete diffusion. *arXiv preprint arXiv:2509.25171*, 2025c.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Sophia Vincoff, Oscar Davis, Ismail Ilkan Ceylan, Alexander Tong, Joey Bose, and Pranam Chatterjee. SOAPIA: Siamese-guided generation of off target-avoiding protein interactions with high target affinity. In *ICML 2025 Workshop on Scaling Up Intervention Models*, 2025. URL https://openreview.net/forum?id=j0OpIG7leX.

Ning-Ning Wang, Jie Dong, Yin-Hua Deng, Min-Feng Zhu, Ming Wen, Zhi-Jiang Yao, Ai-Ping Lu, Jian-Bing Wang, and Dong-Sheng Cao. Adme properties evaluation in drug discovery: prediction of caco-2 cell permeability using a combination of nsga-ii and boosting. *Journal of chemical information and modeling*, 56(4):763–773, 2016.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yinuo Zhang, Sophia Tang, Tong Chen, Elizabeth Mahood, Sophia Vincoff, and Pranam Chatterjee. Peptiverse: A unified platform for therapeutic peptide property prediction. *bioRxiv*, pp. 2025–12, 2026.

Yizhen Zheng, Huan Yee Koh, Maddie Yang, Li Li, Lauren T May, Geoffrey I Webb, Shirui Pan, and George Church. Large language models in drug discovery and development: From disease mechanisms to clinical trials. *arXiv preprint arXiv:2409.04481*, 2024.

APPENDIX

A HARD PREDICTOR MODELS

All hard-predictor objectives used in this work are computed using models provided by PeptiVerse (Zhang et al., 2026). For objectives where a hard predictor is available, we use the corresponding PeptiVerse predictor to score candidate peptides and to provide the objective values $s_n(x)$ used by LaPep during guidance and evaluation. When we report results for solubility and half-life without an external evaluation platform, the reported metrics are computed directly by these PeptiVerse predictors. We keep the predictor implementations and scoring protocols fixed across all experimental settings to ensure that differences across guidance sources are attributable to the guidance mechanism rather than changes in the underlying evaluators.

B LANGUAGE MODEL IMPLEMENTATION DETAILS

B.1 MODEL CHOICE AND CONFIGURATION

For our baselines, the language-based components in our system are implemented using QWEN3, a recent family of causal language models designed to support both high-quality instruction following and explicit internal reasoning. Unless otherwise stated, we use QWEN3-0.6B, a dense 0.6B-parameter model, as the default backbone. This model offers a favorable balance between computational efficiency and reasoning capability, while maintaining strong performance in multi-turn dialogue, code generation, and logical inference.

In selected experiments requiring increased representational capacity, we additionally evaluate QWEN3-14B. The architectural interface and inference pipeline remain identical across model sizes, ensuring controlled comparisons.

B.2 CONTEXT LENGTH AND GENERATION BUDGET

Qwen3 models support a maximum context length of up to 32,768 tokens. For all experiments reported in the main paper, we allow a maximum of 5,000 newly generated tokens per model invocation. This limit was chosen to provide sufficient headroom for long-horizon reasoning while avoiding unnecessary generation overhead.

B.3 REASONING MODE CONTROL

Qwen3 uniquely supports explicit switching between *thinking* and *non-thinking* modes within a single model. When thinking mode is enabled, the model generates an internal reasoning trace enclosed within a `<think>...</think>` block, followed by the final response. This mode is particularly useful for tasks involving multi-step logical reasoning, mathematical derivations, or structured planning.

By contrast, disabling thinking mode suppresses the generation of intermediate reasoning content and yields behavior comparable to standard instruction-tuned models. We employ this mode in latency-sensitive or high-throughput settings.

In multi-turn interactions, we follow best practices by excluding the reasoning trace from the conversational history. Only the final response is retained for subsequent turns, preventing unnecessary context growth and reducing the risk of compounding errors.

B.4 SAMPLING PARAMETERS

Unless otherwise specified, we use the following decoding parameters:

Thinking mode. Temperature $= 0.6$, top-$p = 0.95$, top-$k = 20$, and minimum probability $= 0$. Greedy decoding is explicitly avoided, as it consistently degrades reasoning quality and may lead to repetitive outputs.

Non-thinking mode. Temperature $= 0.7$, top-$p = 0.8$, top-$k = 20$, and minimum probability $= 0$.

To further mitigate degenerate repetitions in long-form generations, we optionally apply a presence penalty of up to 1.5. Higher values were found to occasionally reduce linguistic coherence and are therefore not used.

Table S1: **Ablation of guidance sources for solubility and half-life optimization in LaPep.** We report the average solubility score and predicted half-life in hours for peptides generated under three settings: hard predictor guidance only, language model guidance only, and unconditional generation. For these two metrics, no external evaluator is used and all reported values are computed directly by the corresponding hard predictors. The best metrics are highlighted in bold.

Guidance	Solubility	Half-Life (h)
Hard Predictor Only	**0.653**	**2.4135**
Language Model only	0.5915	1.6762
Unconditional	0.6508	1.5199

Table S2: **Language model choice does not improve solubility optimization in LaPep.** We report the average solubility score for peptides generated using language-model-only guidance with Qwen3, Llama 3, and Kimi K2, and compare against hard predictor guidance and an unconditional baseline. All solubility values are computed by the same hard predictor. The best metric is highlighted in bold.

Guidance	Solubility
Qwen3	0.5915
Llama 3	0.6011
Kimi K2	0.6222
Hard Predictor	**0.6530**
Unconditional	0.6508

B.5 Prompting and Output Standardization

For benchmarking and evaluation, we standardize prompts to encourage consistent output structure. In mathematical reasoning tasks, prompts explicitly request step-by-step reasoning and require the final answer to be enclosed in a marked delimiter. For multiple-choice questions, we enforce a constrained output format to simplify automated parsing.

When reasoning mode is enabled, outputs are post-processed by separating the internal reasoning trace from the final response. Only the final response is used for evaluation, logging, and downstream processing.

B.6 Agentic and Tool-Augmented Usage

For experiments involving tool use or external function calls, we integrate Qwen3 through a standardized agent framework that encapsulates tool-calling templates and parsers. This abstraction allows the model to invoke external tools (e.g., code execution or retrieval modules) while maintaining consistent prompting and decoding behavior.

All agentic experiments are served through an OpenAI-compatible API interface, enabling seamless integration with existing tooling and evaluation pipelines.

B.7 Reproducibility

All prompts, decoding parameters, and post-processing steps are fixed across runs. Where applicable, random seeds are controlled at the framework level. We release all prompt templates and output-parsing utilities to ensure full reproducibility of the reported results.

Table S3: **Language model scale does not reliably improve half-life optimization in LaPep.** We report the average predicted half-life in hours for peptides generated using language-model-only guidance with Qwen3 at three model sizes, and compare against hard predictor guidance and an unconditional baseline. All half-life values are computed by the same hard predictor. The best metric is highlighted in bold.

Model Size	Half-Life (h)
Qwen3 0.6B	1.6762
Qwen3 14B	1.9302
Qwen3 32B	1.5988
Hard Predictor	**2.4135**
Unconditional	1.5199

Prompt

```
You are an expert in peptide design and {objective} optimization.

# Task
Given the peptide sequence (tokens are single characters from a fixed vocabulary): {sequence}

We will edit exactly ONE position (0-indexed, i.e., the first position is 0) to improve {objective}.
Target position (0-indexed): {position}
Current token at that position is: {current_token}

Your job:
1) Propose candidate replacement tokens for position {position} that are likely to improve {objective}.
2) Also estimate the likelihood that keeping the token unchanged is best (i.e., no substitution improves {objective}).

# Allowed replacement tokens
You MUST choose replacements ONLY from this allowed set (do not output anything else):
L A G V S E R T I D P K Q N F Y M H W C Ŷ Ĉ Ĝ Ė K Ṃ Ł Ċ Ṯ

# Constraints
- Do NOT propose the same token as the current token ({current_token}) as a substitution.
- Propose exactly 5 unique candidate tokens, each starting with the word "Substitution" followed by the token, representing
a different possible substitution at the specified position.
- Also output exactly 1 "Stay unchanged" line.
- For each line, provide a confidence score (a decimal between 0 and 1) representing how likely you believe the substitution
(or staying unchanged) will improve {objective}. These confidence scores will be used to guide further design steps.
- Sort the 5 substitutions by confidence from highest to lowest.
- Output must contain ONLY the 6 lines in the exact format below. No extra text.
- Do not always suggest only the first tokens in the allowed set. Consider all allowed tokens equally. Your substitutions
should include tokens from across the entire allowed set, not just those at the beginning.

# Output format (EXACT)
Stay unchanged, Confidence <score>
Substitution <TOKEN>, Confidence <score>
Substitution <TOKEN>, Confidence <score>
Substitution <TOKEN>, Confidence <score>
Substitution <TOKEN>, Confidence <score>
Substitution <TOKEN>, Confidence <score>
```

Figure S1: **LaPep Prompt Template.** Fixed natural-language prompt used to interface with large language models (Qwen, Kimi K2, and LLaMA) during sampling. The prompt provides the current peptide sequence, a selected mutation position, and a target design objective, and instructs the language model to propose a small set of candidate replacement tokens along with confidence scores. These proposals define a restricted local neighborhood that is evaluated and filtered by hard property predictors during guided sampling. Holding the prompt template fixed across models enables controlled comparison of language model behavior.