Generating Structurally Diverse Therapeutic Peptides with GFlowNet

Edward Wijaya
StemRIM, Inc.
wijaya@stemrim.com

Abstract

Reinforcement learning approaches for therapeutic peptide generation suffer from mode collapse, converging to narrow regions of sequence space even when explicit diversity penalties are applied. Fine-grained analysis reveals persistent mode-seeking behavior invisible to standard diversity metrics.

We propose GFlowNet for peptide generation, which samples sequences proportionally to reward rather than maximizing expected reward. This objective provides diversity through proportional sampling without requiring explicit output diversity penalties. Comparing against GRPO with explicit diversity enforcement, GFlowNet achieves substantially more uniform sequence sampling and fewer repetitive motifs. Critically, when diversity mechanisms are removed from the reward, GRPO collapses completely while GFlowNet maintains natural diversity. These results demonstrate that proportional sampling is inherently robust to reward function design, offering a key advantage for drug discovery pipelines requiring diverse candidates.

1 Introduction

Therapeutic peptides represent a rapidly growing class of drug candidates, offering advantages over small molecules in specificity and over biologics in manufacturing cost [Muttenthaler et al., 2021]. Computational peptide design has emerged as a promising approach to accelerate drug discovery [Strokach et al., 2020], but clinical success depends critically on *diversity*: drug discovery programs require structurally distinct candidates for robust patent portfolios, comprehensive validation, and hedging against failure modes.

Reinforcement learning (RL) provides an effective framework for goal-directed sequence generation, optimizing policies to maximize expected reward $\mathbb{E}[R(x)]$ [Angermueller et al., 2020]. However, RL methods suffer from **mode collapse**: by optimizing for expected reward, they converge to narrow regions of sequence space, producing limited diversity even with explicit diversity penalties [Jain et al., 2022]. Standard mitigations such as diversity penalties address symptoms rather than causes. Fine-grained analysis reveals that even when coarse metrics appear satisfactory, RL methods exhibit systematic preferences for particular motifs that become catastrophic when diversity mechanisms are weakened.

We propose using Generative Flow Networks (GFlowNet) for therapeutic peptide generation [Bengio et al., 2021]. Unlike reward-maximizing methods, GFlowNet samples sequences *proportionally* to their reward:

$$P(x) \propto R(x) \tag{1}$$

This objective explores the reward landscape proportionally rather than concentrating on modes, providing diversity through the sampling distribution without requiring explicit output diversity penalties.

Preprint.

We present the first comparison of GFlowNet versus GRPO with diversity penalty (GRPO-D) for peptide generation. Our contributions:

- **Fine-grained diversity analysis**: Metrics beyond coarse sequence identity (dipeptide concentration, reward variance, consecutive repeats) that reveal hidden mode-seeking behavior.
- **Robustness characterization**: Systematic evaluation across reward configurations shows GFlowNet maintains diversity where GRPO-D collapses. With entropy-gated rewards, GFlowNet achieves $5.4\times$ more uniform dipeptide sampling; without entropy gating, GRPO-D exhibits complete collapse while GFlowNet maintains natural diversity.

2 Related Work

RL for Sequence Generation. Policy gradient methods have been applied to molecule design [Olivecrona et al., 2017] and protein optimization [Angermueller et al., 2020]. Group Relative Policy Optimization (GRPO) improves memory efficiency by eliminating the critic [Shao et al., 2024]. However, reward-maximizing approaches are prone to mode collapse [Jain et al., 2022], which diversity penalties mitigate but do not eliminate.

GFlowNet. GFlowNets [Bengio et al., 2021] learn to sample proportionally to reward. Trajectory balance [Malkin et al., 2022] enables stable training, while extensions include continuous domains [Lahlou et al., 2023] and local credit assignment [Pan et al., 2023]. Applications span molecular design [Bengio et al., 2021], Bayesian structure learning [Deleu et al., 2022], and multi-objective optimization [Jain et al., 2023].

Protein and Peptide Generation. Diffusion models have achieved success in structure-based protein design [Watson et al., 2023], while protein language models enable sequence generation [Madani et al., 2023]. For therapeutic peptides, diversity of generated candidates remains underexplored [Muttenthaler et al., 2021].

Diversity in Generative Models. Quality-diversity algorithms like MAP-Elites [Mouret and Clune, 2015] maintain diverse archives but require behavior descriptors. Diversity penalties [Shao et al., 2024] add explicit terms to the objective to enforce output diversity, which oppose the reward-maximizing objective. GFlowNet's proportional sampling objective naturally provides diversity through the sampling distribution without requiring explicit output diversity penalties, though like other RL methods it uses policy entropy regularization for exploration [Haarnoja et al., 2018].

3 Methodology

3.1 Problem Formulation

We formulate therapeutic peptide generation as a sequential decision-making problem. Let $\mathcal{A} = \{A_1, \ldots, A_{20}\}$ denote the 20 standard amino acids, and let $x = (a_1, \ldots, a_L)$ be a peptide sequence of length $L \in [10, 30]$. The generation process constructs sequences autoregressively: at each step t, the model selects an action $a_t \in \mathcal{A} \cup \{\texttt{STOP}\}$ given the partial sequence $s_t = (a_1, \ldots, a_{t-1})$. A reward function $R : \mathcal{X} \to \mathbb{R}^+$ assigns a non-negative score to complete sequences, reflecting desirable properties such as stability, binding affinity, and naturalness.

The key distinction between methods lies in their training objective. Reward maximization methods optimize:

$$\max_{\theta} \mathbb{E}_{x \sim \pi_\theta}[R(x)] \tag{2}$$

In contrast, GFlowNet learns a policy π_θ that samples sequences proportionally to their reward:

$$\pi_\theta(x) \propto R(x) \tag{3}$$

3.2 GFlowNet Architecture

Our GFlowNet models the forward policy $P_F(a_t|s_t)$, the probability of selecting amino acid a_t given partial sequence s_t, using a causal Transformer [Vaswani et al., 2017]. The architecture consists of 4

Transformer layers with hidden dimension 256 and 8 attention heads. The vocabulary includes 20 amino acid tokens plus special tokens for sequence start, stop, and padding.

Training Objective. We train using Sub-Trajectory Balance (STB) loss [Madan et al., 2023], a variant of Trajectory Balance [Malkin et al., 2022] that provides improved credit assignment by computing balance constraints on sub-trajectories. STB's per-step credit assignment provides more stable gradient signals during training, particularly beneficial for variable-length sequence generation. For a complete trajectory $\tau = (s_0 \to s_1 \to \cdots \to x)$, the loss aggregates over all sub-trajectories:

$$\mathcal{L}_{\text{STB}} = \sum_{t=1}^{L} \lambda^{L-t} \left(\log Z + \sum_{i=1}^{t} \log P_F(a_i|s_i) - \log R_t - \sum_{i=1}^{t} \log P_B(s_{i-1}|s_i) \right)^2 \tag{4}$$

where Z is a learnable partition function, P_B is the backward policy, R_t is the intermediate reward at step t, and $\lambda = 0.9$ weights later steps more heavily. For autoregressive generation, we use uniform backward transitions $P_B = 1$, as each state has exactly one predecessor.

Partition Function Learning. The log partition function $\log Z$ is learned jointly with the policy. We found that the learning rate for $\log Z$ critically affects training stability: a $10\times$ multiplier relative to the policy learning rate prevents both mode collapse (multiplier too low) and undertraining (multiplier too high); see Appendix F for ablation results. We also apply policy entropy regularization with weight 0.01, a standard RL exploration technique [Haarnoja et al., 2018] that prevents premature convergence by penalizing overconfident policies. This differs fundamentally from output diversity penalties (discussed below for GRPO-D): entropy regularization encourages policy exploration, while output diversity penalties explicitly reward the diversity of generated samples. While the STB loss drives the policy toward approximate proportional sampling, practical success requires balancing the $\log Z$ learning rate to maintain stable training dynamics. The diversity benefits emerge from the mode-covering sampling objective even when exact flow balance is not achieved.

3.3 GRPO-D Baseline

For comparison, we extend Group Relative Policy Optimization (GRPO) [Shao et al., 2024] with an explicit output diversity penalty, which we term GRPO-D (D for Diversity). This extension provides a stronger baseline by addressing GRPO's tendency toward mode collapse, enabling a fairer comparison against GFlowNet's diversity from proportional sampling. GRPO-D uses a pretrained ProtGPT2-distilled model [Ferruz et al., 2022] as its policy backbone with frozen token and positional embeddings, giving it the advantage of pretraining on protein sequences. This design choice strengthens the baseline, testing whether GFlowNet's proportional sampling can outperform a reward-maximizing method even when the latter benefits from transfer learning.

Training Objective. GRPO-D optimizes:

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{x\sim\pi_\theta}[R(x)] + \beta \cdot D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) - \lambda \cdot \text{div}(x) \tag{5}$$

where $\beta = 0.04$ controls KL divergence from a reference policy, and $\lambda = 0.15$ weights an explicit diversity penalty combining amino acid frequency rarity (70%) and mean pairwise Levenshtein distance (30%) within each batch.

Note on Architectural Differences. GFlowNet uses a Transformer trained from scratch, while GRPO-D uses a pretrained ProtGPT2-distilled backbone. This design choice intentionally strengthens the GRPO-D baseline by providing transfer learning from protein sequences, creating a comparison where GFlowNet operates at an architectural disadvantage. That GFlowNet achieves superior diversity metrics despite this disadvantage provides stronger evidence for the benefits of proportional sampling. However, this confounds the effect of training objective (trajectory balance vs. reward maximization) with initialization strategy (random vs. pretrained). Future work could isolate these factors by comparing both methods with identical architectures and initialization, though our results suggest the training objective plays a dominant role given GFlowNet's consistent advantages across multiple reward configurations.

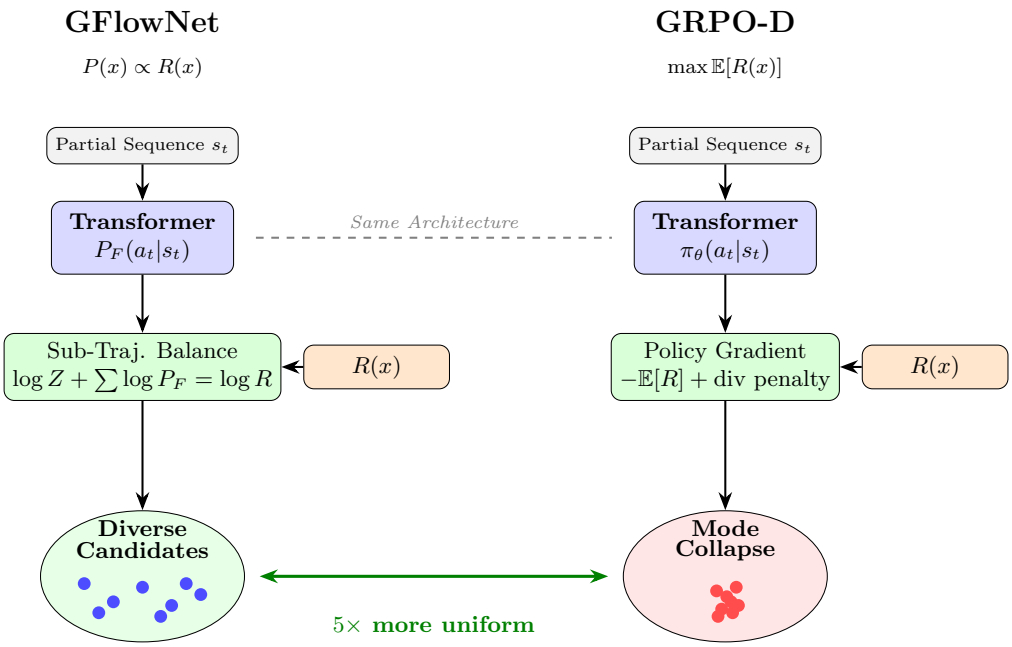

Figure 1: Comparison of GFlowNet and GRPO-D architectures. GFlowNet uses a custom Transformer trained from scratch, while GRPO-D uses a pretrained ProtGPT2-distilled backbone. The methods differ in training objective: GFlowNet learns to sample proportionally to reward ($P(x) \propto R(x)$) via sub-trajectory balance, while GRPO-D maximizes expected reward with a diversity penalty. This difference leads to diverse sampling (GFlowNet) versus mode collapse (GRPO-D).

3.4 Reward Functions

We evaluate both methods under three reward function configurations that progressively test robustness to reward design choices.

ImprovedReward (with entropy gating). This reward combines naturalness scoring with explicit diversity enforcement:

$$R(x) = \text{naturalness}(x) \cdot \text{entropy_gate}(x) \cdot \text{length_gate}(x) \tag{6}$$

The entropy gate returns 0 for sequences with amino acid entropy below a threshold, explicitly penalizing repetitive patterns. Naturalness is computed as $\sigma(\|\boldsymbol{\mu}(x)\|/\tau)$ where $\boldsymbol{\mu}(x)$ is the mean-pooled ESM-2 [Lin et al., 2023] embedding, σ is the sigmoid function, and $\tau = 10$ is a temperature parameter. The embedding norm serves as a proxy for how well the sequence fits ESM-2's learned representation space: natural proteins have consistent embedding magnitudes while degenerate sequences exhibit abnormal norms.

CompositeReward (without entropy gating). This reward removes explicit diversity enforcement:

$$R(x) = \text{stability}(x) \cdot \text{binding}(x) \cdot \text{naturalness}(x)^{0.5} \tag{7}$$

Each component is predicted by MLP heads on frozen ESM-2 embeddings. The 0.5 exponent on naturalness creates diminishing returns, preventing sequences that are merely "natural-looking" from scoring highly without also exhibiting good stability and binding properties. Without entropy gating, this configuration tests the robustness of each generation method to reward design choices.

ESM2-PLL (pure pseudo-log-likelihood). This reward uses only ESM-2's masked language model score:

$$R(x) = \exp\left(\frac{1}{L}\sum_{i=1}^{L}\log P_{\text{ESM}}(a_i|x_{-i})\right) \tag{8}$$

This is a known degenerate reward that directly favors repetitive sequences, as repeated amino acids have high conditional probability. We include it to test boundary conditions. Mathematical details of all reward components are provided in Appendix D.

3.5 Evaluation Metrics

We evaluate generated sequences using both coarse and fine-grained metrics.

Coarse Metrics. Standard diversity and quality measures include: (1) *Sequence diversity*: $1 - \bar{d}$, where $\bar{d}$ is the mean pairwise identity computed as the fraction of matching amino acids at corresponding positions, $d(x, y) = |\{i : x_i = y_i\}|/\max(|x|, |y|)$; (2) *Unique ratio*: fraction of distinct sequences in the sample; (3) *Mean/max reward*: average and maximum $R(x)$ across samples.

Fine-Grained Metrics. We introduce metrics that reveal subtle mode-seeking behavior: (1) *Dipeptide concentration*: percentage of all dipeptides belonging to the top-10 most frequent; (2) *Dipeptide entropy*: Shannon entropy of the dipeptide frequency distribution, normalized by $\log_2(400)$ (maximum entropy over 400 possible dipeptides); (3) *Coefficient of variation*: σ/μ of the reward distribution; (4) *Quality floor*: 5th percentile reward; (5) *Consecutive repeats*: percentage of sequences with > 2 identical consecutive amino acids; (6) *Max consecutive repeat*: longest run of identical amino acids in any sequence.

3.6 Implementation Details

Both methods are implemented in PyTorch and trained on a single NVIDIA Tesla T4 GPU. GFlowNet trains for 10,000 steps with batch size 64; GRPO-D trains for 1,000 iterations with the same batch size. We use Adam optimization with learning rate 3×10^{-4}. Reward model predictors were trained on two datasets: the FLIP meltome benchmark [Dallago et al., 2021] for stability prediction ($\sim$28,000 protein sequences) and the PepBDB database [Wen et al., 2019] for binding prediction ($\sim$4,800 peptide-protein complexes). For evaluation, we sample 1,000 sequences from each trained model. Complete hyperparameters and dataset details are provided in Appendix C and E.

4 Experiments

We design experiments to answer two research questions: (1) Do GFlowNet and GRPO-D differ on fine-grained diversity metrics when coarse metrics appear equivalent? (2) How do the methods compare when diversity-enforcing mechanisms are removed from the reward?

4.1 Experimental Setup

We train both GFlowNet and GRPO-D to generate therapeutic peptides of length 10–30 amino acids optimized for stability and naturalness, sampling 1,000 sequences from each trained model for evaluation. GFlowNet uses a custom Transformer trained from scratch, while GRPO-D uses a pretrained ProtGPT2-distilled backbone, giving it the advantage of transfer learning from protein sequence data. We evaluate under two reward configurations: ImprovedReward (with entropy gating) and CompositeReward (without entropy gating), testing both normal operation and robustness to reward design.

4.2 Main Comparison: With Entropy Gating

Table 1 presents coarse diversity and quality metrics for both methods under ImprovedReward.

At the coarse level, both methods appear equivalent: sequence diversity is nearly identical (0.945 vs 0.948), and both achieve 100% unique sequences. GFlowNet achieves slightly higher mean reward (0.630 vs 0.623) despite not maximizing reward directly, demonstrating that proportional

Table 1: Coarse metrics under ImprovedReward. Both methods achieve similar diversity (∼0.95) and quality at this level of analysis.

Metric	GFlowNet	GRPO-D
Sequence Diversity	0.945	0.948
Mean Reward	**0.630**	0.623
Max Reward	0.643	**0.645**
Unique Ratio	1.00	1.00

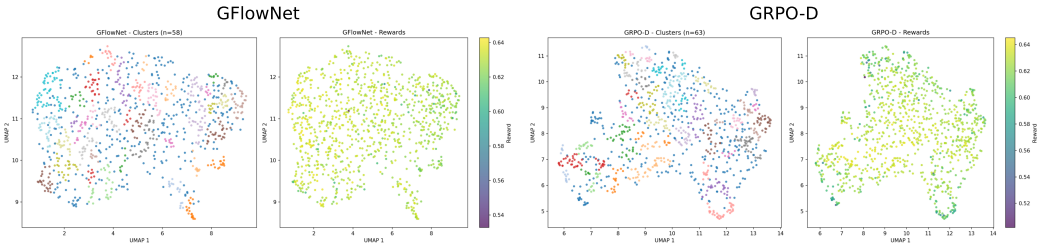

Figure 2: UMAP visualization of ESM-2 embeddings. Left: GFlowNet samples. Right: GRPO-D samples. Both methods achieve broad coverage, but fine-grained analysis reveals differences in local structure.

sampling does not sacrifice reward quality for diversity. Figure 2 shows UMAP projections of ESM-2 embeddings for samples from both methods, confirming comparable coverage of the embedding space.

4.3 Fine-Grained Analysis

We compare GFlowNet against GRPO-D (with diversity penalty λ=0.15) as this represents the strongest RL baseline; vanilla GRPO (λ=0) exhibits mode collapse even with entropy gating, as shown in Table 3. Table 2 reveals substantial differences invisible to coarse metrics.

Figure 3 visualizes the dipeptide concentration difference. GRPO-D's top-10 dipeptides account for 21.7% of all dipeptides, compared to only 4.0% for GFlowNet. This $5.4\times$ concentration difference indicates that GRPO-D samples preferentially favor certain sequence motifs, even when overall diversity metrics appear healthy.

GFlowNet also produces more consistent quality: its coefficient of variation is $1.9\times$ lower (0.012 vs 0.023), and its quality floor (5th percentile reward) is 3.6% higher (Figure 4). This means GFlowNet's worst samples are substantially better than GRPO-D's worst samples, while maintaining comparable peak performance.

4.4 Ablation: Removing Diversity Mechanisms

To test robustness to reward design, we systematically remove diversity enforcement mechanisms from GRPO and measure the effect. Table 3 presents results for two ablations: (1) removing entropy gating from the reward function, and (2) removing the diversity penalty from the training objective.

Without entropy gating. When we replace ImprovedReward with CompositeReward (removing the entropy gate), GRPO-D exhibits **complete mode collapse**: 100% of its 1,000 samples contain the tripeptide pattern `RMMRMMRMM`, with dipeptides RM, MR, and MM accounting for 92.4% of all dipeptides. In contrast, GFlowNet maintains natural diversity (0.937) with no degenerate patterns.

Without diversity penalty. When we remove the diversity penalty (λ=0) while keeping entropy gating, vanilla GRPO exhibits severe motif collapse: its top-10 dipeptides account for 52.5% of all dipeptides, $13\times$ worse than GFlowNet. While coarse metrics (0.863 diversity, 97.5% unique) appear acceptable, the extreme dipeptide concentration reveals collapse to a small set of preferred motifs.

Table 2: Fine-grained metrics under ImprovedReward. GFlowNet shows substantially more uniform sampling and fewer degenerate patterns. Bold indicates better.

Metric	GFlowNet	GRPO-D	Ratio
Sampling Uniformity			
Top-10 dipeptide concentration	**4.0%**	21.7%	5.4×
Dipeptide entropy (normalized)	**0.98**	0.84	1.16×
Reward Consistency			
Coefficient of variation	**0.012**	0.023	1.9×
5th percentile (quality floor)	**0.619**	0.597	+3.6%
Sequence Quality			
% with consecutive repeats (>2)	**5.9%**	23.1%	3.9×
Max consecutive repeat	**4**	8	2×

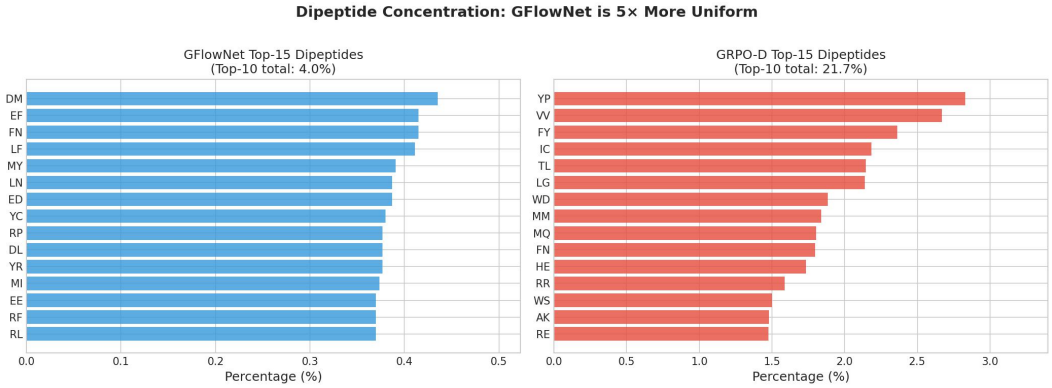

Figure 3: Dipeptide frequency distribution. GFlowNet (left) shows near-uniform sampling across dipeptides, while GRPO-D (right) concentrates on specific motifs.

These ablations demonstrate that RL methods require diversity enforcement at **two levels**:

1. **Reward function**: Entropy gating prevents complete collapse (Ablation 1)
2. **Training objective**: Diversity penalty λ prevents motif collapse (Ablation 2)

GFlowNet requires neither mechanism; its proportional sampling objective provides robust diversity without explicit output diversity enforcement.

4.5 Reward Function Robustness

Table 4 summarizes method performance across all configurations tested. Note that the GRPO λ setting only affects GRPO; GFlowNet has no diversity penalty parameter.

Across all configurations, GFlowNet either wins or ties; it never underperforms GRPO-D. The ESM2-PLL result (both methods collapse) demonstrates that proportional sampling provides additional but not unlimited robustness: pathological reward functions that directly incentivize degenerate sequences defeat both approaches.

5 Discussion

Our experiments demonstrate that GFlowNet consistently outperforms GRPO-D on diversity metrics, but the practical implications extend beyond benchmark numbers. We analyze why proportional sampling provides diversity through its mode-covering objective and examine the broader implications for reward function design.

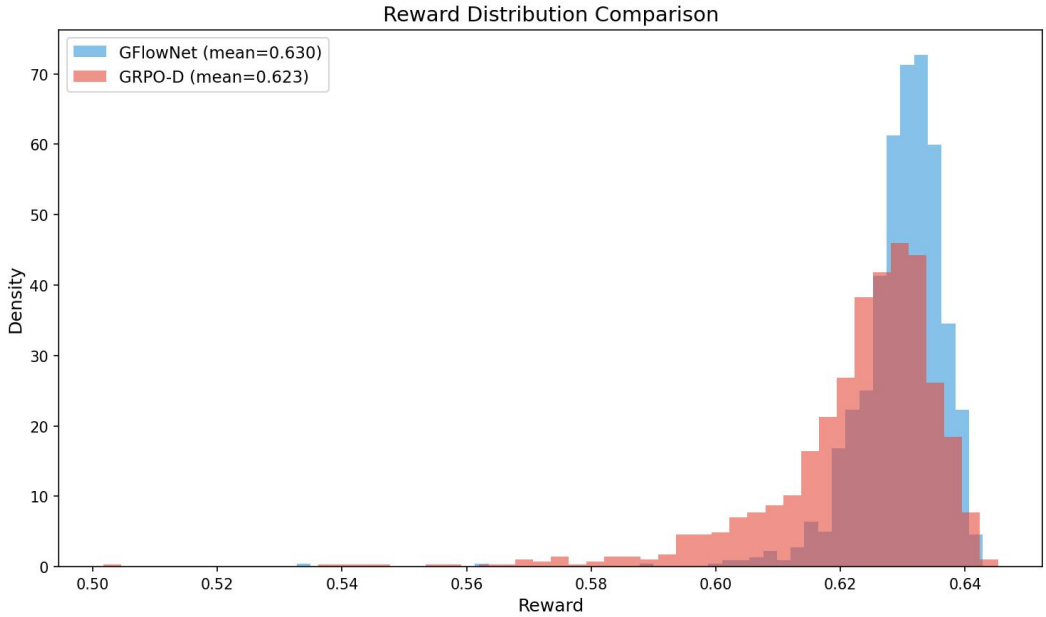

Figure 4: Reward distributions. GFlowNet shows tighter concentration around high rewards with a higher quality floor.

Table 3: Ablation studies. GRPO requires explicit diversity enforcement at both reward AND training objective levels to avoid mode collapse; GFlowNet requires neither.

Configuration	Seq. Diversity	Dipeptide Conc.	Unique Ratio
Ablation 1: Remove entropy gating from reward (CompositeReward)			
GFlowNet	**0.937**	**1.8%**	**100%**
GRPO-D (λ=0.15)	0.501	92.4%	74%
Ablation 2: Remove diversity penalty from training (ImprovedReward)			
GFlowNet	**0.945**	**4.0%**	**100%**
Vanilla GRPO (λ=0)	0.863	52.5%	97.5%

5.1 Why Proportional Sampling Maintains Diversity

The fundamental difference between GFlowNet and reward-maximizing methods can be understood through their optimization objectives and their effect on the learned distribution.

Mode-seeking versus mode-covering. Reward maximization and proportional sampling represent two fundamentally different ways of relating a learned distribution π_θ to a target reward landscape $R(x)$.

Reward maximization optimizes:

$$\max_\theta \mathbb{E}_{x\sim\pi_\theta}[R(x)] = \max_\theta \sum_x \pi_\theta(x)R(x) \tag{9}$$

This objective is maximized when π_θ places all probability mass on $x^* = \arg\max_x R(x)$. Even with entropy regularization or diversity penalties, the gradient signal consistently pushes probability mass *toward* high-reward modes. Diversity mechanisms resist this push but do not eliminate it; they create an equilibrium between mode-seeking and diversity-preserving forces, which can be disrupted by changes to hyperparameters or reward design.

Table 4: Robustness across configurations. GFlowNet maintains diversity regardless of reward function; GRPO requires both entropy gating in the reward AND diversity penalty (λ) in training.

		Sequence Diversity	
Reward Function	GRPO λ	GFlowNet	GRPO
ImprovedReward	0.15	0.945	0.948
ImprovedReward	0 (vanilla)	0.945	0.863
CompositeReward	0.15	**0.937**	0.501
ESM2-PLL (degenerate)	0.15	0.312	0.298

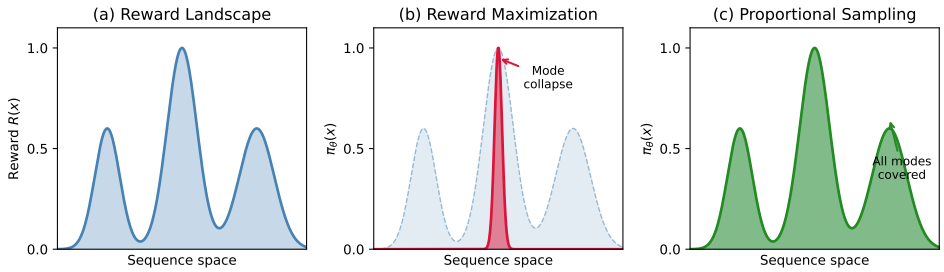

Figure 5: Mode-seeking versus mode-covering behavior. **Left:** A multimodal reward landscape with three modes of varying height. **Center:** Reward maximization concentrates probability on the global maximum (mode-seeking). **Right:** Proportional sampling allocates probability across all modes according to reward magnitude (mode-covering).

In contrast, GFlowNet's sub-trajectory balance objective trains the policy to satisfy:

$$\pi_\theta(x) = \frac{R(x)}{Z} \tag{10}$$

This objective performs *distributional matching* rather than optimizing expected reward. The policy learns to *cover* the reward landscape proportionally rather than *seek* its modes. There is no force pushing probability mass toward modes beyond what the reward itself specifies.

Figure 5 illustrates this distinction. Given a multimodal reward landscape, reward maximization concentrates on the global maximum, while proportional sampling allocates probability mass across all modes according to their reward magnitude.

Information-theoretic interpretation. The proportional sampling distribution $P(x) \propto R(x)$ can be derived as the solution to a maximum entropy problem. Consider finding the distribution that achieves a target expected log-reward while maximizing entropy:

$$\max_P H(P) \quad \text{subject to} \quad \mathbb{E}_P[\log R(x)] = c \tag{11}$$

The solution is the exponential family distribution $P(x) \propto R(x)$, which is precisely GFlowNet's target. This reveals that proportional sampling is the *least committed* distribution consistent with a given reward structure; it makes no assumptions beyond what the reward specifies, preserving maximum uncertainty (and thus diversity) about which sequences to generate.

In contrast, reward maximization implicitly minimizes entropy by concentrating probability mass. Even entropy-regularized RL:

$$\max_\theta \mathbb{E}_{\pi_\theta}[R(x)] + \alpha H(\pi_\theta) \tag{12}$$

only *slows* entropy collapse rather than preventing it. The regularization coefficient α must be carefully tuned, and the optimal α depends on the reward scale and landscape shape, information rarely available a priori.

Why diversity penalties are insufficient. Our ablation studies reveal that GRPO requires diversity enforcement at two levels: entropy gating in the reward function *and* explicit diversity penalty (λ) in the training objective. Removing either causes collapse (Table 3).

This dual requirement reflects a fundamental tension: diversity penalties *oppose* the optimization objective rather than *changing* it. The policy gradient still points toward reward modes; the diversity penalty provides a counterforce. This creates several failure modes:

1. **Hyperparameter sensitivity**: The penalty weight λ must balance mode-seeking against diversity. Too low permits collapse; too high sacrifices reward quality.
2. **Batch-dependent gradients**: Diversity penalties computed within batches provide noisy gradient estimates that may not reflect population-level diversity.
3. **Metric-scale mismatch**: GRPO-D's diversity penalty (mean pairwise Levenshtein distance) operates at the sequence level, as does the standard evaluation metric (mean pairwise identity). Both measure global dissimilarity between sequences but are blind to local structure. This alignment explains why GRPO-D matches GFlowNet on coarse diversity (0.948 vs 0.945) while exhibiting $5.4\times$ worse dipeptide concentration (21.7% vs 4.0%); the policy satisfies the sequence-level penalty while concentrating on repeated local motifs.

Multi-scale diversity from distributional matching. GFlowNet's proportional sampling objective contains no explicit diversity metric, which paradoxically makes it robust across all scales. Because the policy learns to match the reward distribution rather than optimize a specific diversity measure, there is no metric that can be "gamed" at one scale while violated at another. The diversity emerges naturally from covering the reward landscape, providing consistent behavior from sequence-level statistics down to local motif distributions.

5.2 The Partition Function as Diversity Regulator

A unique aspect of GFlowNet training is the learned partition function Z. Our $\log Z$ ablation (Appendix F) reveals that this scalar parameter plays a critical role in maintaining diversity.

When $\log Z$ is learned too slowly ($5\times$ multiplier), the policy sharpens faster than the normalization can track, creating a runaway positive feedback loop that causes mode collapse. When learned too quickly (15–20$\times$), over-normalization diffuses gradients and prevents reward learning entirely.

At the optimal rate ($10\times$), the partition function acts as an automatic diversity regulator: as the policy begins concentrating on high-reward sequences, Z increases correspondingly, reducing the effective gradient magnitude and preventing further sharpening. This self-correcting mechanism has no analog in reward-maximizing methods.

5.3 Structural Hedging in Drug Discovery

Reward functions in computational drug discovery capture known desirable properties such as stability, binding affinity, and naturalness, but cannot fully encode pharmacokinetics, tissue penetration, immunogenicity, or off-target toxicity. When candidates fail clinical trials for reasons outside the reward model, structurally similar candidates often fail for the same reasons.

GFlowNet's mode-covering behavior produces structurally distinct families of candidates. If one family fails due to an unforeseen property (e.g., poor plasma stability, inability to cross the blood-brain barrier), other families with different structural characteristics may succeed. This "structural hedging" transforms a fragile single-candidate pipeline into a resilient multi-family portfolio, particularly valuable when the same mechanism of action applies across multiple therapeutic indications.

Figure 7 (Appendix G) illustrates this portfolio effect in biochemical property space. Comparing GFlowNet against vanilla GRPO ($\lambda = 0$), GFlowNet samples span all four therapeutic quadrants, from cell-penetrating peptides (hydrophilic, cationic) to stable long-acting candidates (hydrophilic, anionic), while GRPO-vanilla collapses to a single narrow property band.

5.4 Limitations

Our analysis has several limitations that suggest directions for future work.

Task specificity. We evaluate on a single therapeutic peptide generation task. While the theoretical arguments for GFlowNet's diversity advantages are general, empirical validation across other sequence generation domains (molecules, DNA, longer proteins) would strengthen these conclusions.

Reward function coverage. We test three reward configurations, but the space of possible reward functions is vast. Characterizing precisely which reward properties lead to GFlowNet success versus failure remains an open question.

Computational considerations. GFlowNet trains from scratch while GRPO-D benefits from pretrained weights. A controlled comparison with equivalent pretraining would isolate the effect of the training objective from architectural advantages. The optimal $\log Z$ learning rate may also vary across tasks, requiring some hyperparameter search.

6 Conclusion

Proportional sampling fundamentally outperforms reward maximization for therapeutic peptide generation. By learning to sample sequences proportionally to reward, GFlowNet achieves diversity through its sampling objective that explicit output diversity penalties (as in our GRPO-D baseline) cannot match. This diversity comes at no cost to reward quality: GFlowNet achieves comparable or higher mean reward than GRPO-D, eliminating the reward-diversity tradeoff inherent in penalty-based approaches.

Our experiments reveal that GFlowNet's advantages extend beyond coarse diversity metrics. Even when both methods achieve similar sequence diversity ($\sim$0.95), GFlowNet produces $5.4\times$ more uniform dipeptide sampling, $1.9\times$ lower reward variance, and $3.9\times$ fewer repetitive sequences. These fine-grained differences become dramatic under stress: when entropy gating is removed from the reward function, GRPO-D exhibits complete mode collapse while GFlowNet maintains natural diversity.

Our study has limitations. The ESM2-PLL experiment demonstrates that proportional sampling provides additional but not unlimited robustness: pathological reward functions that directly incentivize degenerate sequences defeat both approaches. Our comparison uses different architectures (GFlowNet trained from scratch vs. GRPO-D with pretrained weights), which confounds the effect of training objective with initialization strategy. While this strengthens the GRPO-D baseline and demonstrates GFlowNet's advantages despite the disadvantage, it limits our ability to isolate the specific contribution of proportional sampling versus architecture choice. Additionally, our evaluation focuses on a single therapeutic peptide generation task; generalization to other sequence generation domains remains to be validated.

The GFlowNet framework is modular with respect to the reward function. Alternative therapeutic objectives (such as cell proliferation, mobilization efficiency, or solubility) can be incorporated by training appropriate predictor heads on the shared ESM-2 backbone or by integrating external predictors, without modifying the generation architecture.

Future work could extend GFlowNet to multi-objective optimization, where proportional sampling across Pareto frontiers may provide natural trade-off exploration. Incorporating wet-lab feedback through active learning presents another promising direction, leveraging GFlowNet's calibrated sampling to efficiently explore chemical space while minimizing experimental cost.

Broader Impact

This work accelerates therapeutic peptide candidate generation, with implications for drug discovery timelines and costs.

Risks and Limitations. Generated peptides require wet-lab validation; our method accelerates candidate generation but does not replace experimental verification. As with any biological sequence generator, theoretical dual-use concerns exist, though therapeutic peptides have limited misuse potential compared to other biological agents. Our model may also inherit biases from training data, potentially underexploring certain regions of peptide space.

Environmental Considerations. Computational exploration may reduce wet-lab experiments during early discovery, decreasing associated resource consumption.

Acknowledgments

This work was conducted at StemRIM, Inc. The author thanks the company for providing computational resources and supporting this research.

References

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations*, 2020.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27381–27394, 2021.

Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, 2021.

Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pages 518–528. PMLR, 2022.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, 2022. doi: 10.1038/s41467-022-32007-7.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure FP Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, et al. Biological sequence design with GFlowNets. In *International Conference on Machine Learning*, pages 9786–9801. PMLR, 2022.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-García, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective GFlowNets. In *International Conference on Machine Learning*, pages 14631–14653. PMLR, 2023.

Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-García, Léna Néhale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks. In *International Conference on Machine Learning*, pages 18269–18300. PMLR, 2023.

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

Kanika Madan, Jarrid Rector-Brooks, Maksym Korablyov, Emmanuel Bengio, Moksh Jain, Andrei Nica, Tom Bosc, Yoshua Bengio, and Nikolay Malkin. Learning GFlowNets from partial episodes for improved convergence and stability. In *International Conference on Machine Learning*, pages 23467–23483. PMLR, 2023.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41 (8):1099–1106, 2023.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In *Advances in Neural Information Processing Systems*, volume 35, pages 5955–5967, 2022.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Markus Muttenthaler, Glenn F King, David J Adams, and Paul F Alewood. Trends in peptide drug discovery. *Nature Reviews Drug Discovery*, 20(4):309–325, 2021.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):1–14, 2017.

Ling Pan, Nikolay Malkin, Dinghuai Zhang, and Yoshua Bengio. Better training of GFlowNets with local credit and incomplete trajectories. In *International Conference on Machine Learning*, pages 26878–26890. PMLR, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. DeepSeek-Math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Fast and flexible protein design using deep graph neural networks. *Cell Systems*, 11(4):402–411, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Zeyu Wen, Junfeng He, Haiyan Tao, and Sheng-You Huang. Pepbdb: a comprehensive structural database of biological peptide-protein interactions. *Bioinformatics*, 35(1):175–177, 2019.

A Code and Model Availability

To support reproducibility and facilitate future research, we release our complete implementation, trained models, and experimental artifacts.

Source Code. The full codebase, including training scripts, evaluation pipelines, and configuration files, is available at:

`https://github.com/ewijaya/gflownet-peptide`

Pre-trained Models. We provide pre-trained model weights on Hugging Face Hub:

- **GFlowNet Forward Policy**: A causal Transformer trained with sub-trajectory balance loss for peptide generation.
  `https://huggingface.co/littleworth/gflownet-peptide`
- **Stability Predictor**: An ESM-2 based model trained on FLIP Meltome data for thermal stability prediction.
  `https://huggingface.co/littleworth/peptide-stability-predictor`

License. All code and models are released under the MIT License.

B Guidelines for Practitioners

Based on our findings, we offer the following guidelines for choosing between GFlowNet and reward-maximizing methods.

When to prefer GFlowNet.

- **Diverse candidate generation**: When downstream applications require structurally distinct candidates (drug discovery, patent portfolios, hedging against failure modes).
- **Uncertain reward quality**: When the reward function may contain spurious optima or unintended shortcuts. GFlowNet's robustness to reward imperfections (Table 4) provides a safety margin.
- **Avoiding hyperparameter sensitivity**: GFlowNet requires tuning the $\log Z$ learning rate but does not require balancing competing objectives (reward vs. diversity penalty).

When reward maximization may suffice.

- **Single best candidate**: When only the highest-reward sequence matters, reward maximization is directly aligned with the goal.
- **Well-validated rewards**: When the reward function has been extensively validated and is known to lack degenerate optima.
- **Explicit diversity requirements**: When specific diversity constraints (e.g., minimum edit distance) can be hard-coded into the reward or sampling procedure.

Reward function design. Our experiments suggest principles for reward functions compatible with both methods:

- **Avoid direct repetition incentives**: ESM2-PLL fails because repeated amino acids have high conditional probability under masked language models. Rewards derived from autoregressive or masked probabilities may inadvertently favor degenerate sequences.
- **Multiplicative composition**: Our successful reward functions (ImprovedReward, CompositeReward) use multiplicative combination, ensuring all components must be satisfied. Additive rewards allow compensation, potentially enabling reward hacking.
- **Test with ablations**: Before deployment, test reward robustness by removing diversity-enforcing components. If generation collapses, the reward contains implicit assumptions that may not hold in all settings.

C Hyperparameters

Table 5 lists the hyperparameters used for both methods.

Table 5: Hyperparameters for GFlowNet and GRPO-D training.

Hyperparameter	GFlowNet	GRPO-D
Architecture		
Policy backbone	Custom Transformer	ProtGPT2-distilled
Transformer layers	4	pretrained
Hidden dimension	256	pretrained
Attention heads	8	pretrained
Vocabulary size	23 (20 AA + 3 special)	pretrained
Frozen embeddings	—	Yes (token + positional)
Value head hidden dim	—	256
Training		
Learning rate	3×10^{-4}	3×10^{-4}
Batch size	64	64
Training steps/iterations	10,000	1,000
Optimizer	Adam	Adam
Method-specific		
Loss function	Sub-Trajectory Balance	PPO-style
STB decay (λ_{sub})	0.9	—
$\log Z$ learning rate multiplier	$10\times$	—
Entropy regularization weight	0.01	—
KL coefficient (β)	—	0.04
Diversity penalty weight (λ)	—	0.15
Generation		
Min sequence length	10	10
Max sequence length	30	30
Evaluation samples	1,000	1,000
Compute		
GPU	Tesla T4	Tesla T4
Training time	∼5–6 hours	∼10 hours
Random seeds	42, 123, 456	42, 123, 456

Architecture Selection. We conducted a scaling ablation comparing GFlowNet architectures from 3.2M to 25M parameters (varying depth 4–8 layers and width 256–512). The baseline 3.2M model achieved the best performance (mean reward 0.977, diversity 0.945), with larger models showing slight degradation: 6.3M deeper (0.971, 0.928), 12.6M wider (0.951, 0.907). The 25M scaled model required more conservative hyperparameters to prevent mode collapse and consequently underfit. These results suggest the task is reward-limited rather than capacity-limited for peptide lengths 10–30, justifying our selection of the computationally efficient 3.2M architecture.

D Reward Function Design

This appendix details the mathematical formulation and design rationale for each reward component. Throughout this section, σ denotes the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{13}$$

which maps any real value to the interval $(0, 1)$.

D.1 Multiplicative Reward Structure

Our reward functions use multiplicative composition:

$$R(x) = \prod_i f_i(x) \tag{14}$$

where each $f_i : \mathcal{X} \to [0, 1]$. This ensures all components must be satisfied simultaneously; if any component approaches zero, the total reward approaches zero, preventing reward hacking on individual dimensions.

D.2 Naturalness

Naturalness measures how well a sequence fits the learned representation space of natural proteins:

$$\text{naturalness}(x) = \sigma\left(\frac{\|\boldsymbol{\mu}(x)\|}{\tau}\right) \tag{15}$$

where $\boldsymbol{\mu}(x) = \frac{1}{L}\sum_{i=1}^{L} \mathbf{h}_i$ is the mean-pooled ESM-2 embedding (excluding special tokens), $\|\cdot\|$ denotes L2 norm, and $\tau = 10.0$ is a temperature parameter.

Intuition: Natural proteins occupy consistent regions in embedding space with characteristic embedding magnitudes. Degenerate sequences (e.g., homopolymers) produce abnormal embedding norms, yielding low naturalness scores.

D.3 Entropy Gate

The entropy gate penalizes sequences with low amino acid diversity:

$$\text{entropy_gate}(x) = \sigma\left(k \cdot (H_{\text{norm}}(x) - \theta)\right) \tag{16}$$

where the normalized Shannon entropy is:

$$H_{\text{norm}}(x) = \frac{H(x)}{\log_2 20}, \quad H(x) = -\sum_{i=1}^{20} p_i \log_2 p_i \tag{17}$$

with p_i being the frequency of amino acid i in sequence x. The hyperparameters are threshold $\theta = 0.5$ and sharpness $k = 10.0$.

Intuition: Homopolymers (e.g., MMMM...) have $H = 0$, yielding gate values near zero. The threshold requires sequences to use at least a moderate diversity of amino acids, preventing mode collapse to repetitive patterns.

D.4 Length Gate

The length gate ensures minimum sequence length:

$$\text{length_gate}(x) = \sigma\left(k \cdot (L - L_{\min})\right) \tag{18}$$

where $L = |x|$ is sequence length, $L_{\min} = 10$, and $k = 0.5$.

Intuition: Peptides shorter than approximately 10 amino acids are unlikely to adopt stable secondary structures. The sigmoid provides a soft constraint that maintains gradient flow near the boundary.

D.5 Stability and Binding Predictors

Both stability and binding are predicted using MLP heads on frozen ESM-2 embeddings with identical architecture but separately trained weights:

$$\text{stability}(x) = \sigma(\text{MLP}_{\text{stab}}(\boldsymbol{\mu}(x))) \tag{19}$$

$$\text{binding}(x) = \sigma(\text{MLP}_{\text{bind}}(\boldsymbol{\mu}(x))) \tag{20}$$

where each MLP has architecture $\mathbb{R}^d \to [256] \to [128] \to [1]$ with ReLU activations and dropout 0.1.

Stability: MLP_{stab} is trained on experimental thermal stability data from the FLIP meltome benchmark to predict thermodynamic stability (related to $\Delta G_{\text{folding}}$).

Binding: MLP_{bind} is trained on peptide-protein binding data from PepBDB to classify sequences as binders versus non-binders. This formulation predicts general binding propensity without incorporating specific target protein information.

E Datasets

This section describes the datasets used for training the reward model predictors.

FLIP Meltome. We use the meltome split from the FLIP benchmark [Dallago et al., 2021] for training the stability predictor. This dataset contains approximately 28,000 protein sequences with experimentally measured melting temperatures (in °C). While the dataset contains full-length proteins (median length 413 amino acids), the ESM-2 embeddings generalize to peptide-length sequences, enabling stability prediction for our 10–30 amino acid generated peptides. We use the standard train/validation/test splits provided by the benchmark.

PepBDB. For binding prediction, we use peptide sequences from the PepBDB database [Wen et al., 2019], which contains approximately 4,800 unique peptides (10–50 amino acids) extracted from experimentally resolved peptide-protein complex structures in the Protein Data Bank. Since all peptides in this dataset are verified structural binders, we use binary classification labels. The dataset provides diverse peptide sequences spanning multiple binding modalities and target protein families.

ESM-2 Backbone. Both predictors use frozen ESM-2 embeddings (esm2_t33_650M_UR50D) as input features. ESM-2 was pretrained on UniRef50, providing rich sequence representations that transfer effectively to downstream prediction tasks.

ProtGPT2-distilled. The GRPO-D baseline uses the ProtGPT2-distilled-medium model (`littleworth/protgpt2-distilled-medium` on HuggingFace) as its policy backbone. This model is a distilled variant of ProtGPT2 [Ferruz et al., 2022], a GPT-2 based language model pretrained on millions of protein sequences from UniRef50. The distilled version maintains strong performance while reducing computational requirements. We freeze the token and positional embeddings to preserve pretrained sequence knowledge while adding a trainable value head for advantage estimation in the GRPO algorithm.

F Log Z Learning Rate Ablation

The $\log Z$ learning rate multiplier exhibits a sharp Goldilocks zone around $10\times$ the base learning rate. Figure 6 shows training dynamics across four multiplier values, revealing three distinct regimes:

Mode collapse (5×): When the multiplier is too low, the partition function Z learns too slowly to track the policy's sharpening. This creates a runaway feedback loop: the policy concentrates probability mass on high-reward sequences, but $\log Z$ underestimates the true partition function, providing artificially large gradients that further sharpen the policy. The result is catastrophic mode collapse; the unique ratio crashes to near zero as the model generates identical sequences.

Success (10×): At the optimal multiplier, $\log Z$ tracks the policy's evolution, maintaining correct normalization throughout training. The model achieves high mean reward (0.98) while preserving diversity (0.94 sequence diversity, 100% unique sequences).

Undertrained (15–20×): When the multiplier is too high, $\log Z$ overshoots, over-normalizing the reward signal. This diffuses gradients excessively, and the policy fails to learn reward structure, defaulting to near-uniform sampling with low mean reward ($\sim$0.45) despite high diversity.

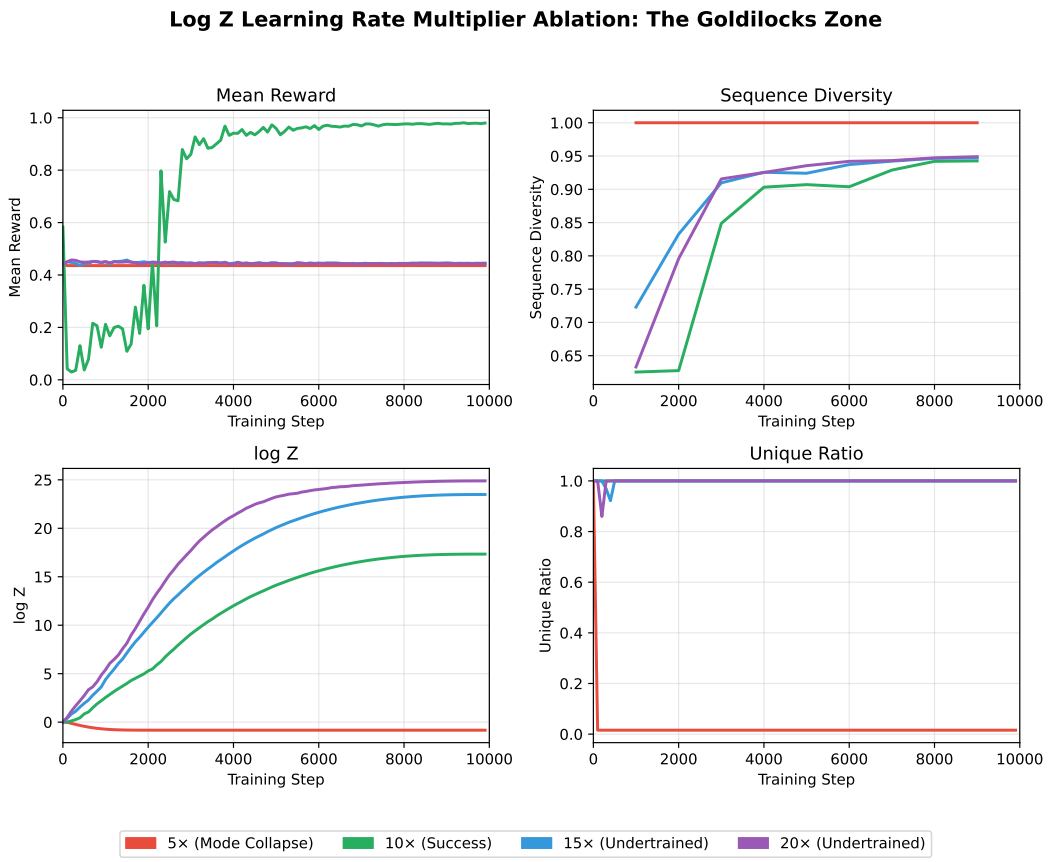

Figure 6: $\log Z$ learning rate ablation showing the Goldilocks zone. **5**× (red): Mode collapse; $\log Z$ stays near zero, unique ratio crashes to zero. **10**× (green): Success; balanced dynamics achieve high reward (0.98) with diversity (0.94). **15–20**× (blue/purple): Undertrained; $\log Z$ grows too fast, reward learning stalls at $\sim$0.45.

G Biochemical Property Space Analysis

To demonstrate the therapeutic relevance of GFlowNet's diversity advantage, we analyze samples in a two-dimensional biochemical property space defined by interpretable physicochemical properties: hydrophobicity (GRAVY) and net charge at physiological pH.

Property definitions. **GRAVY** (Grand Average of Hydropathy) measures overall hydrophobicity using the Kyte-Doolittle scale, computed as the mean hydropathy index across all residues. Positive values indicate hydrophobic sequences; negative values indicate hydrophilic sequences. **Net charge at pH 7** sums the charges of ionizable residues (K, R: $+1$; H: $+0.1$; D, E: -1), indicating the electrostatic character of the peptide.

Therapeutic relevance. These two properties define four quadrants with distinct therapeutic profiles:

- **Hydrophilic + Cationic** (top-left): Cell-penetrating peptides (CPPs) that cross membranes while maintaining solubility. Applications include intracellular delivery and gene therapy.
- **Hydrophobic + Cationic** (top-right): Antimicrobial peptides (AMPs) that disrupt bacterial membranes. Applications include antimicrobials and membrane-active oncology agents.
- **Hydrophilic + Anionic** (bottom-left): Stable, soluble peptides with lower immunogenicity and longer half-life. Suitable for chronic dosing and IV formulations.

- **Hydrophobic + Anionic** (bottom-right): Specialized peptides for lipid-associated targets and membrane anchoring.

Results. Figure 7 compares GFlowNet and vanilla GRPO ($\lambda = 0$) in this property space. GFlowNet samples span all four quadrants, while vanilla GRPO collapses to a single region: 68% of GRPO samples fall in the hydrophilic-cationic quadrant (CPP-like), leaving the anionic half of property space largely unexplored. GFlowNet generates $3\times$ more anionic peptides (44% vs. 14%) and covers the strongly anionic region (charge < -2) that GRPO essentially never reaches (18% vs. $<1\%$).

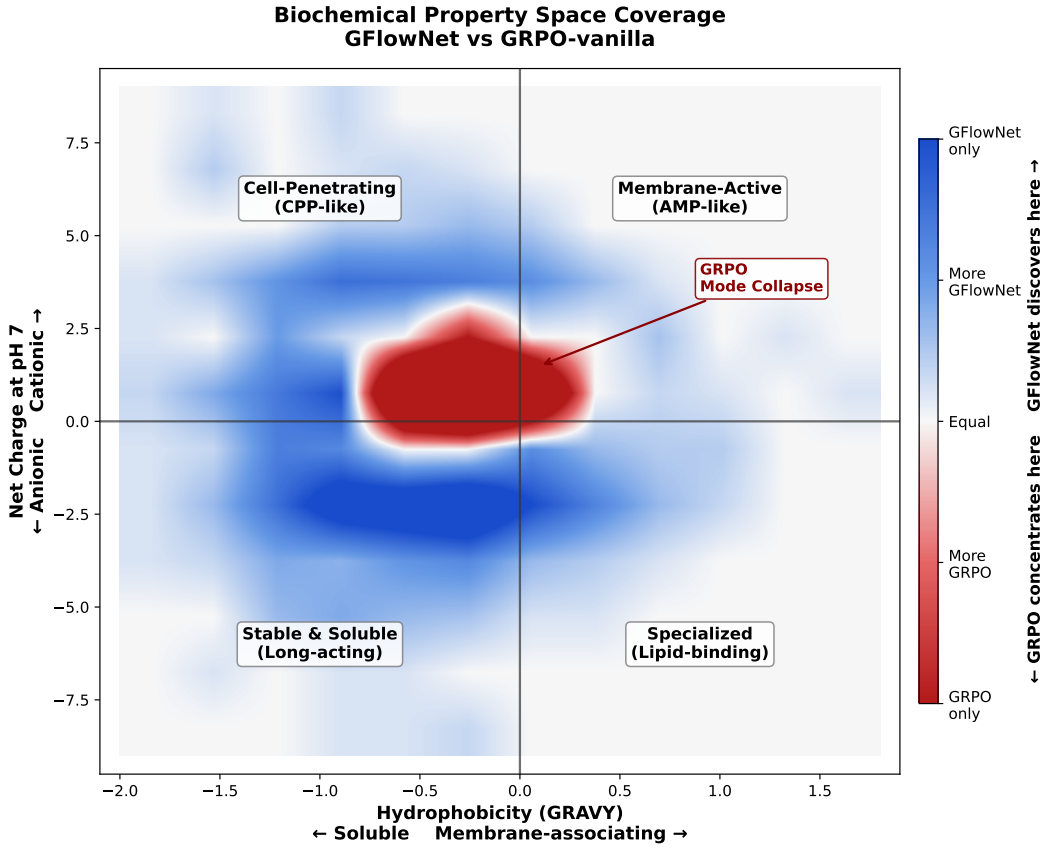

Figure 7: Biochemical property space coverage. Blue regions indicate where GFlowNet discovers more peptides; red indicates where GRPO-vanilla concentrates. The white boundary shows the transition zone. GFlowNet covers all four therapeutic quadrants while GRPO-vanilla exhibits mode collapse to a single property band.

Implications for drug discovery. This property-level diversity provides "portfolio insurance" against development failures. If candidates from one quadrant fail due to unforeseen issues (e.g., poor solubility, immunogenicity, inadequate tissue penetration), structurally distinct candidates from other quadrants remain available. GRPO-vanilla's concentration in a single property band means that a scaffold-level liability could invalidate the entire candidate pool, requiring discovery to restart from scratch.