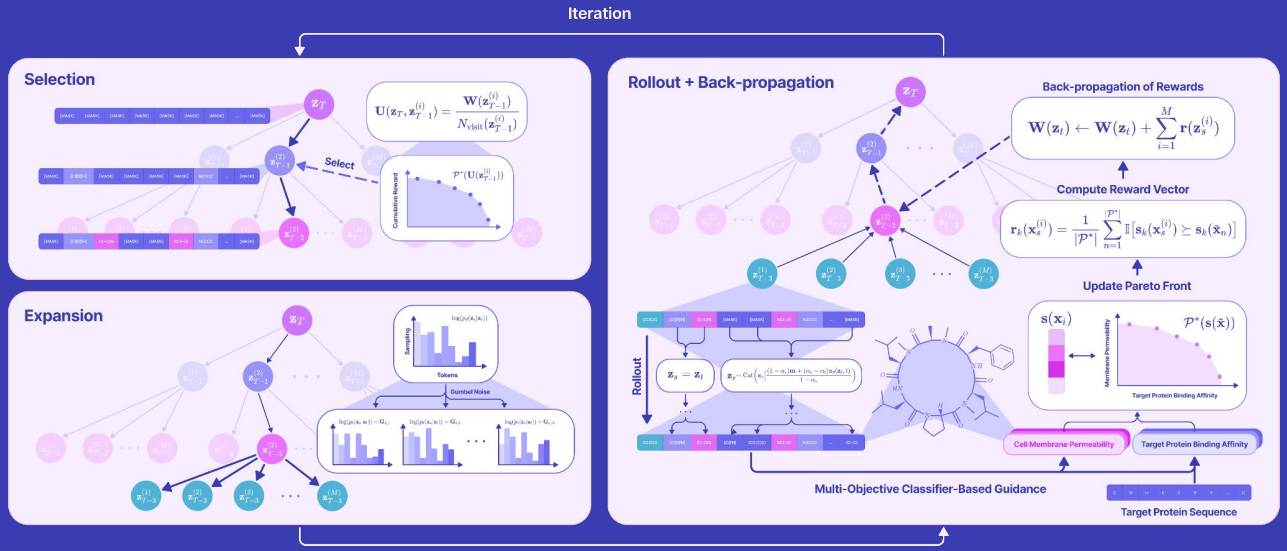
PepTune: *De Novo* Generation of Therapeutic Peptides with Multi-Objective-Guided Discrete Diffusion

Sophia Tang* • Yinuo Zhang* • Pranam Chatterjee



Discrete Diffusion Reading Group | Nov 24, 2025

PepTune Paper

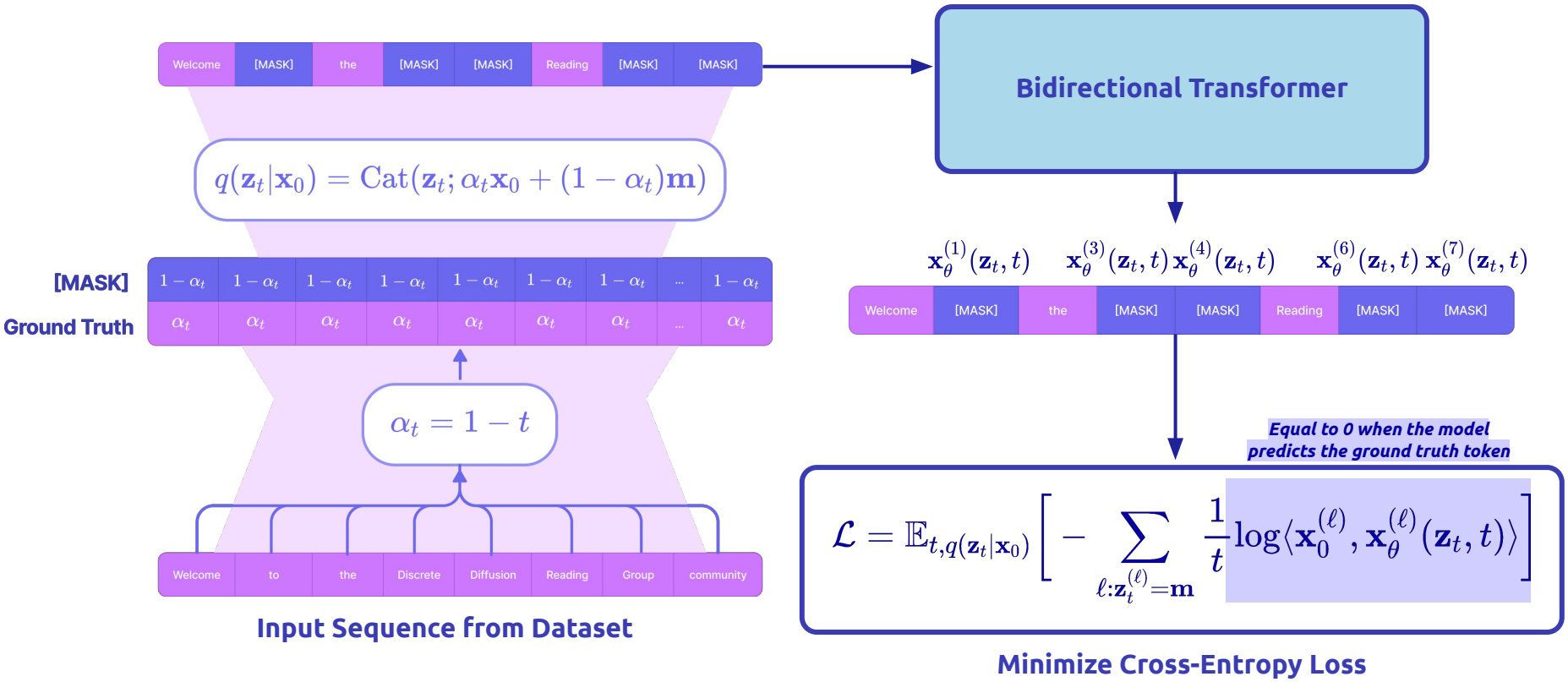
Agenda

1. Brief Primer of Discrete Diffusion
2. Controllability in Unmasking Time
3. Multi-Objective Guidance Discrete Diffusion
4. What’s Next
5. Conclusion + Q&A

Masked Discrete Diffusion

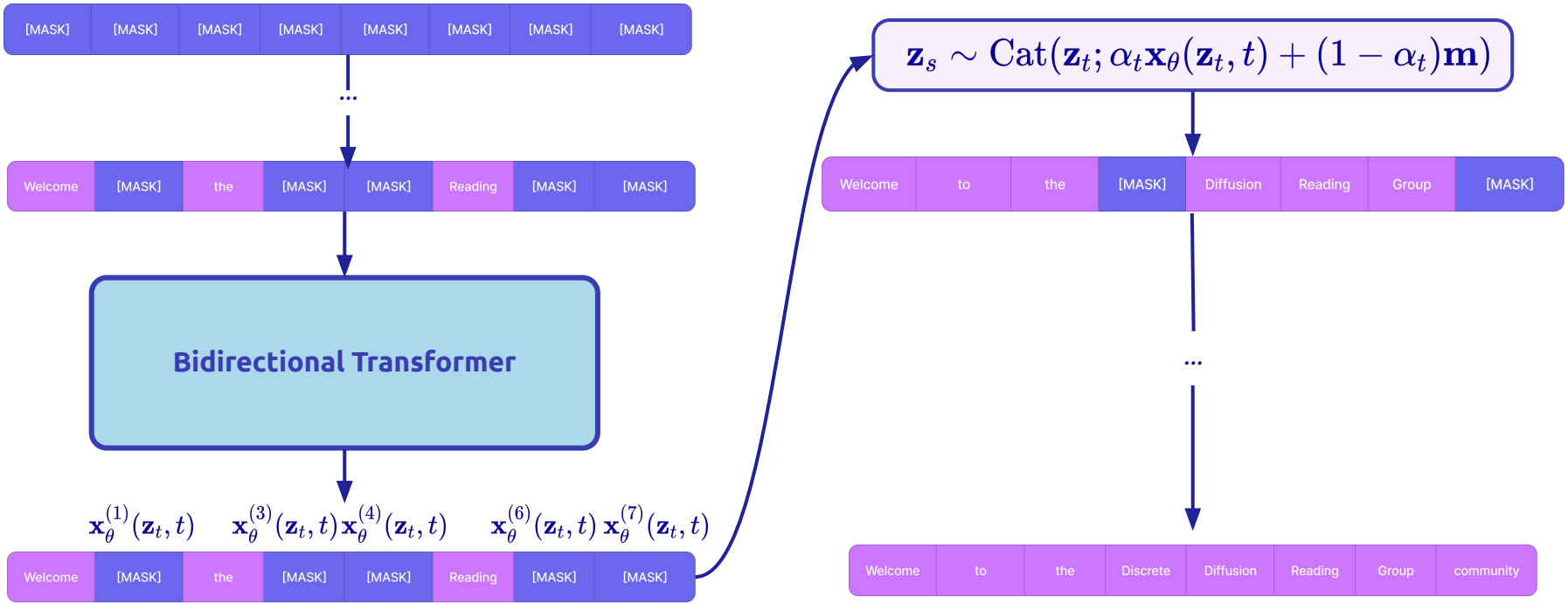
Training Discrete Diffusion

Training a parameterized model to unmask tokens and recover data sequences



Inference Discrete Diffusion

Sampling with the trained model



Discrete Diffusion is Suited for Biological Design

1. **Biological sequences naturally contain long-range, bidirectional dependencies**
2. **Diffusion avoids “positional bias” inherent in autoregressive models**
3. **Discrete diffusion leaves room for controllability in the unmasking order** (as discussed later)

Motivation Behind PepTune

Can we generate valid therapeutic peptides simultaneously optimized across *multiple* properties?

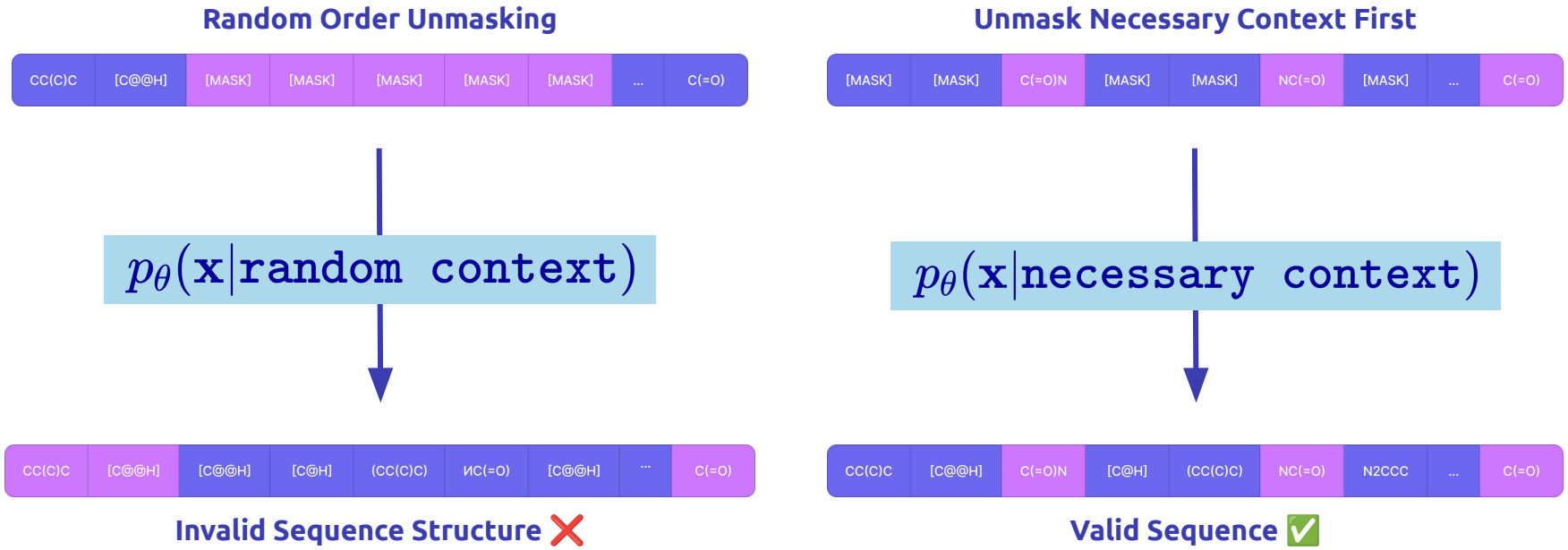
Lack of multi-objective guidance strategies in discrete state spaces

Previous discrete guidance methods rely on projecting to and from the continuous latent space or gradient estimation. Multi-objective guidance strictly in the discrete state space remains underexplored.

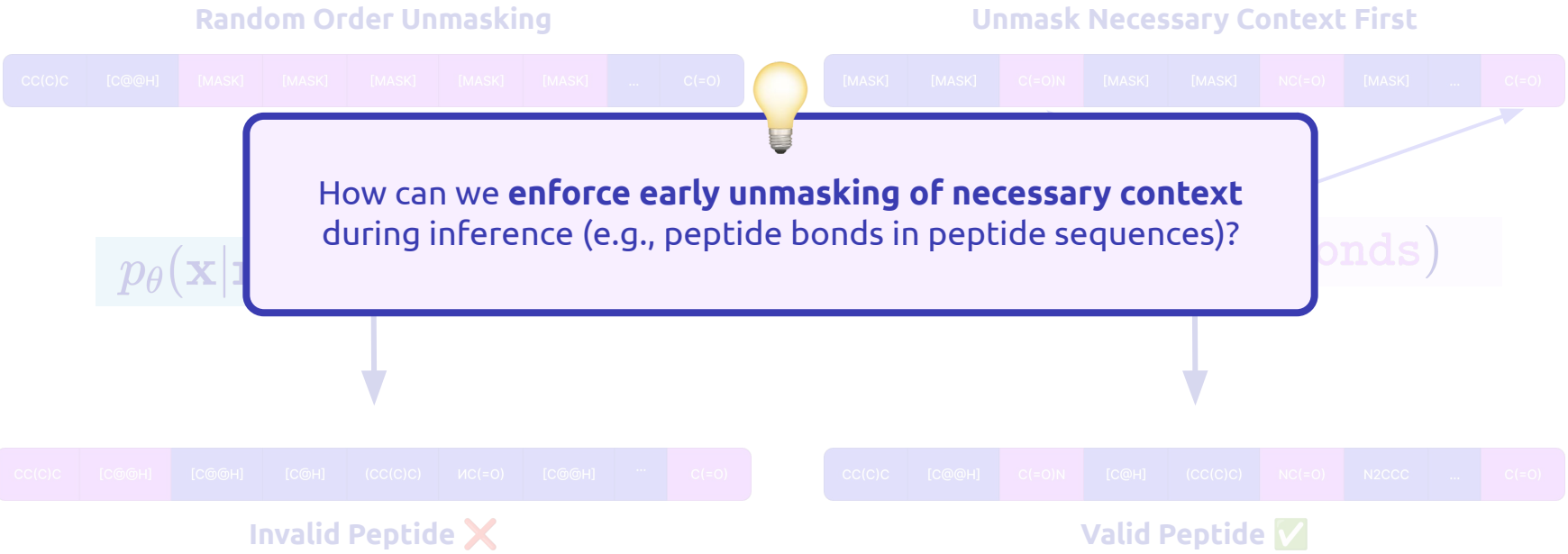
Lack of generative models non-natural and cyclic peptides

Existing sequence-based models represent peptides as sequences of the 20 natural amino acids, but fail to represent diverse space of non-natural amino acids and cyclic peptides with improved therapeutic properties.

Random Unmasking Order Fails to Preserve Key Sequence Structure

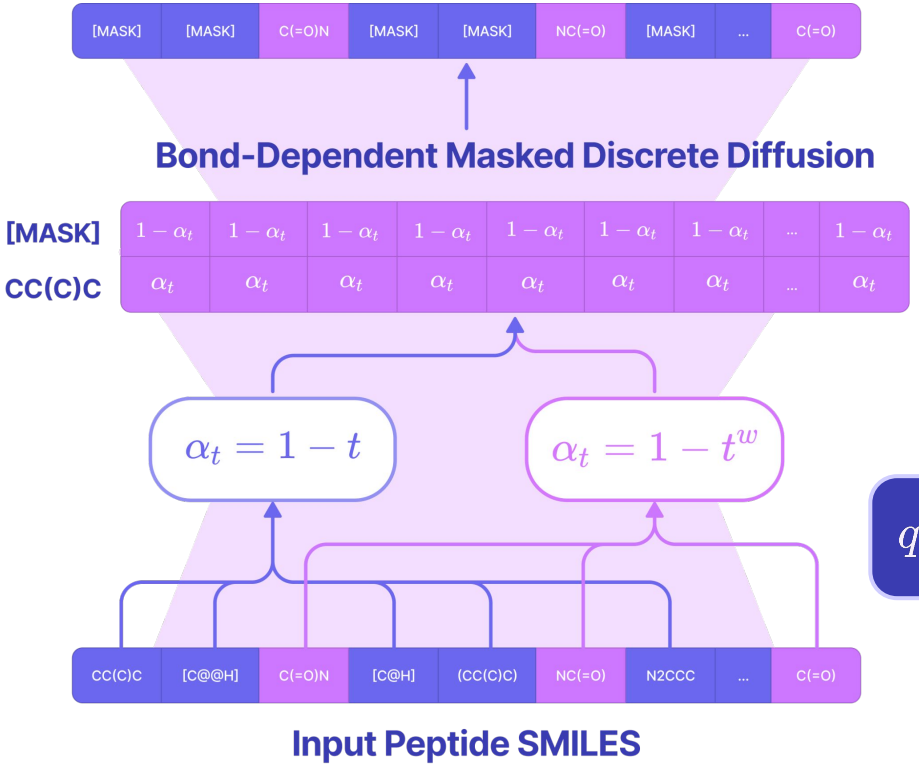


Random Unmasking Order Fails to Preserve Key Sequence Structure



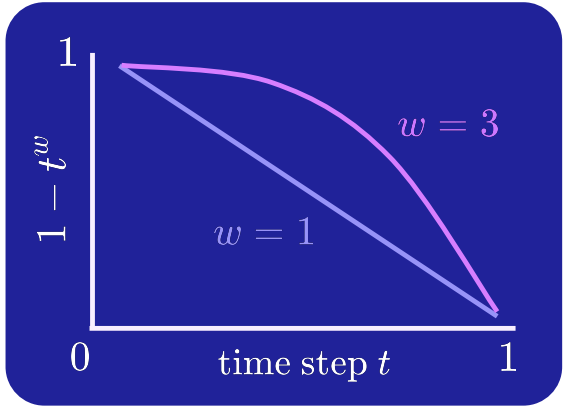
Bond-Dependent Masking Schedule

How do we encourage accurate unmasking of peptide bond tokens?



Late masking of peptide bond tokens

Enforces early unmasking of peptide bond tokens



$$q(\mathbf{z}_t|\mathbf{x}_0) = \text{Cat}(\mathbf{z}_t; \alpha_t(\mathbf{x}_0)\mathbf{x}_0 + (1 - \alpha_t(\mathbf{x}_0))\mathbf{m})$$

$$\alpha_t(\mathbf{x}_0) = \begin{cases} 1 - t^w & \mathbf{x}_0 = \mathbf{b} \\ 1 - t & \mathbf{x}_0 \neq \mathbf{b} \end{cases}$$

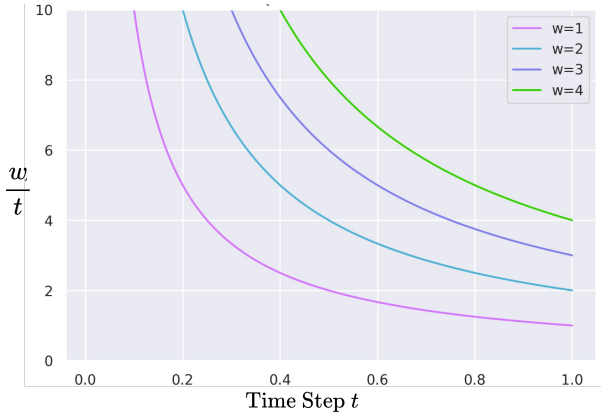
Shi et al. 2024

Continuous-Time Bond-Dependent Loss

How do we encourage accurate unmasking of peptide bond tokens?

Cross-entropy loss of peptide bond tokens

$$\sum_{\ell:\tilde{\boldsymbol{x}}_t^\ell=\boldsymbol{M},\boldsymbol{x}^\ell=\boldsymbol{b}} -\frac{w}{t}\log p^{u_\theta}(\tilde{\boldsymbol{x}}_t)_{\ell,\boldsymbol{x}^\ell}$$



Loss is weighted higher for larger weight as time → 0

Cross-entropy loss of remaining tokens

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(0,1),q_t(\tilde{\boldsymbol{x}}_t|\boldsymbol{x})}\left[\sum_{\ell:\tilde{\boldsymbol{x}}_t^\ell=\boldsymbol{M},\boldsymbol{x}^\ell=\boldsymbol{b}} -\frac{w}{t}\log p^{u_\theta}(\tilde{\boldsymbol{x}}_t)_{\ell,\boldsymbol{x}^\ell} + \sum_{\ell:\tilde{\boldsymbol{x}}_t^\ell=\boldsymbol{M},\boldsymbol{x}^\ell\neq\boldsymbol{b}} -\frac{1}{t}\log p^{u_\theta}(\tilde{\boldsymbol{x}}_t)_{\ell,\boldsymbol{x}^\ell}\right]$$

Continuous-time Bond-Dependent Loss

Bond-Dependent Reverse Posterior

How do we encourage accurate unmasking of peptide bond tokens?

Reverse Transition Probability When Ground Truth Token is a Peptide Bond

$$q(\mathbf{z}_s|\mathbf{z}_t = \mathbf{m}, \mathbf{x}_0 = \mathbf{b}) = \left(1 - \left(\frac{s}{t}\right)^w\right)\mathbf{x}_0 + \left(\frac{s}{t}\right)^w \mathbf{m}$$

Peptide bond has larger probability of being unmasked

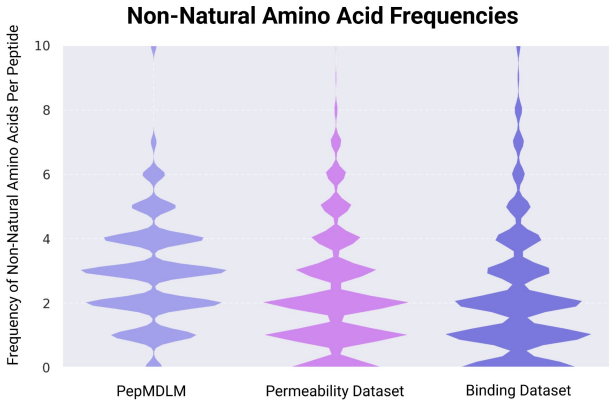
.. and smaller probability of remaining a mask

Reverse Transition Probability When Ground Truth Token is NOT a Peptide Bond

$$q(\mathbf{z}_s|\mathbf{z}_t = \mathbf{m}, \mathbf{x}_0 \neq \mathbf{b}) = \left(1 - \frac{s}{t}\right)\mathbf{x}_0 + \left(\frac{s}{t}\right)\mathbf{m}$$

Generating Diverse Chemically-Modified and Cyclic Peptides

Our model successfully generates non-canonical amino acids with high validity and diversity



Our model expands the search space of therapeutic peptides well beyond any generative model trained on canonical amino acid representations

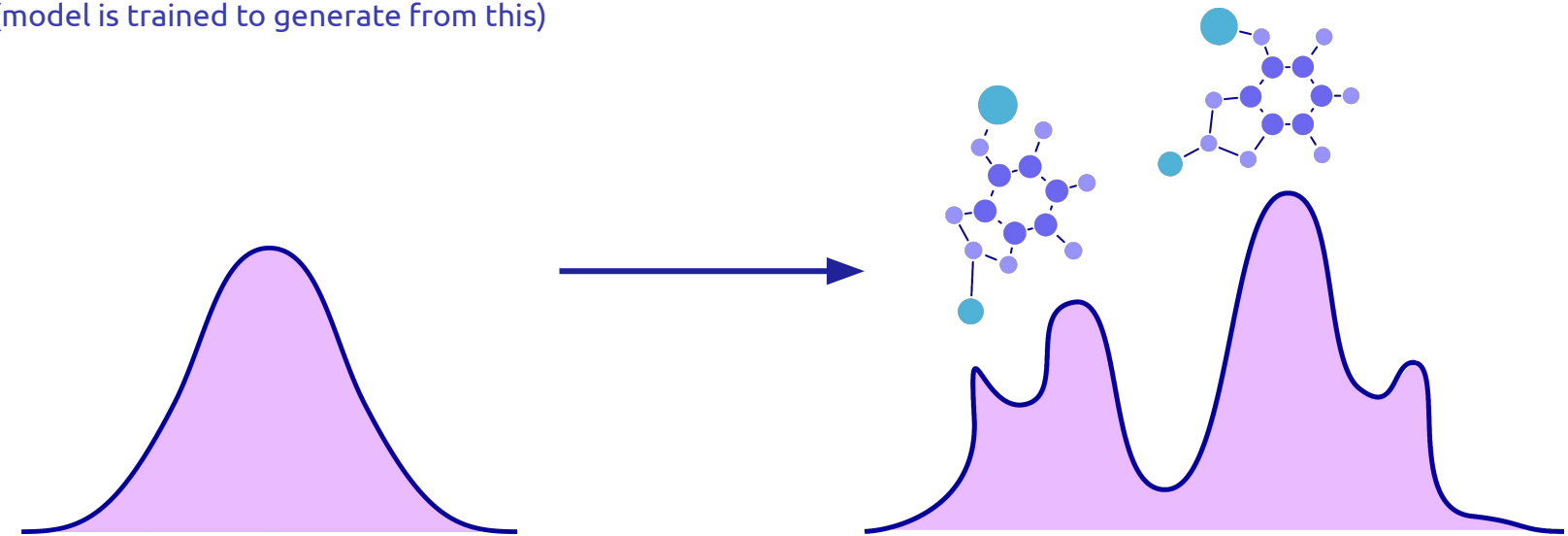
	Permeability Data	Binding Data	PepMDLM
Mean nAAs Per Peptide	2.215	2.150	2.940
Cyclic Peptides (%)	0.467	0.027	0.100

Guidance for Discrete Diffusion

What if we want to sample sequences that are optimized in certain properties?

Data Distribution
(model is trained to generate from this)

Distribution with Property *y*



$$p_{\text{data}}(\mathbf{x})$$

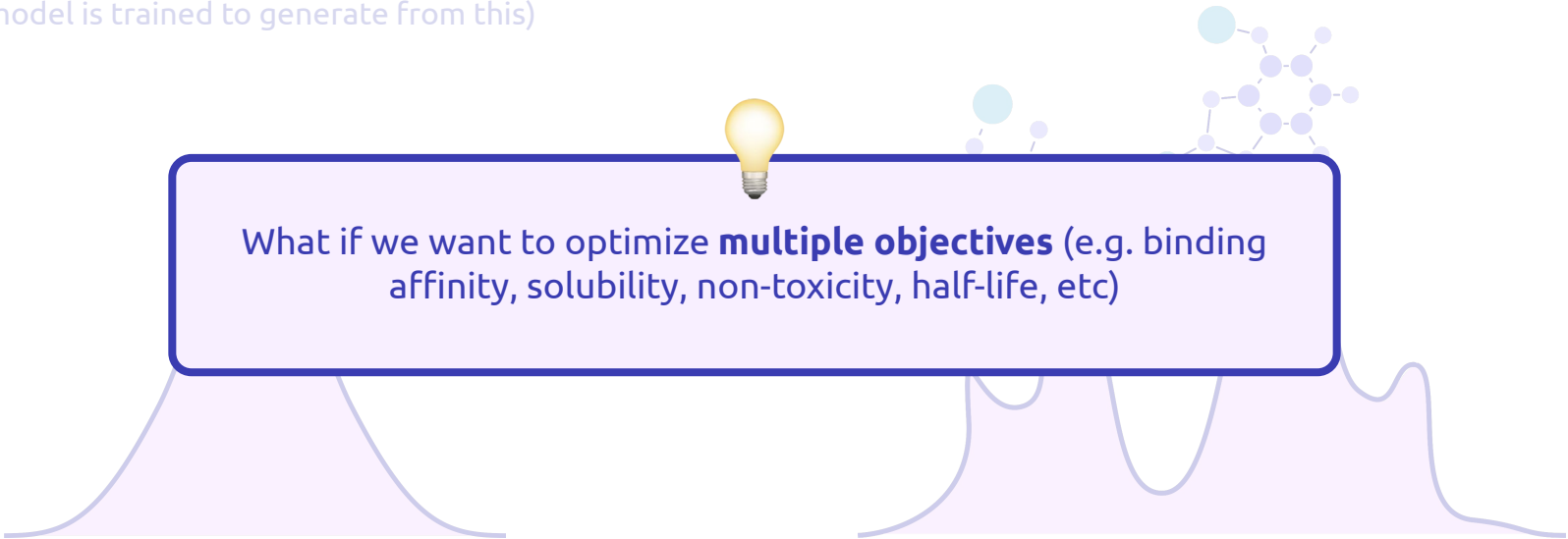
$$p_{\text{target}}(\mathbf{x}|y)$$

Guidance for Discrete Diffusion

What if we want to sample sequences that are optimized in certain properties?

Data Distribution
(model is trained to generate from this)

Distribution with Property *y*



What if we want to optimize **multiple objectives** (e.g. binding affinity, solubility, non-toxicity, half-life, etc)

$p_{\text{data}}(\mathbf{x})$

$p_{\text{target}}(\mathbf{x}|y)$

Multi-Objective Guidance

What does it mean to find the optimal sequences across multiple objectives?

Pareto Dominance

$$\mathbf{s}(\mathbf{x}^\star) \succ \mathbf{s}(\mathbf{x}) \quad \text{iff} \quad \forall k \in [1, K]\ s_k(\mathbf{x}^\star) \geq s_k(\mathbf{x}) \wedge \quad \exists k' \in [1, K]\ s_{k'}(\mathbf{x}^\star) > s_{k'}(\mathbf{x})$$

$\mathbf{x}^\star$ dominates $\mathbf{x}$ — $\mathbf{x}^\star$ is no worse than $\mathbf{x}$ in any objective — $\mathbf{x}^\star$ is strictly better than $\mathbf{x}$ in at least one objective

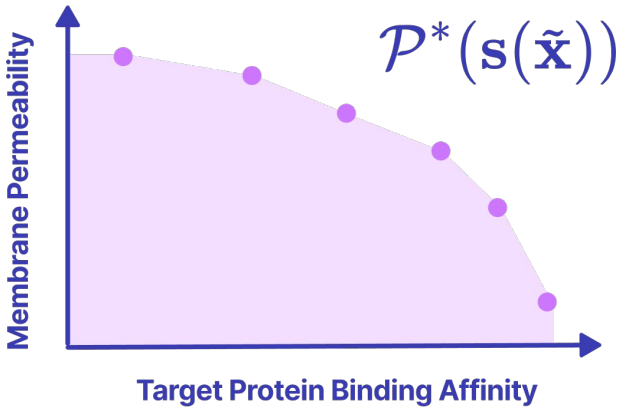
Pareto Non-Dominance

$$\nexists \mathbf{x}^* \in \mathcal{P}^* \ \text{ s.t. } \ \mathbf{s}(\mathbf{x}^*) \succ \mathbf{s}(\mathbf{x})$$

$\mathbf{x}$ is non-dominated

Pareto Optimal Set

$$\mathcal{P}^* = \{(\mathbf{x}, \mathbf{s}(\mathbf{x})) \mid \nexists \mathbf{x}^* \in \mathcal{P}^* \ \text{ s.t. } \ \mathbf{s}(\mathbf{x}^*) \succ \mathbf{s}(\mathbf{x})\}$$



Challenges of Guidance in the Discrete State Space

How do we guide the discrete diffusion process where the gradient is undefined?

Classifier-Based Guidance

$$\nabla_{\mathbf{z}_s} \log p^{\gamma}(\mathbf{z}_s|y, \mathbf{z}_t) = \gamma \nabla_{\mathbf{z}_s} \log p(y|\mathbf{z}_s) + \nabla_{\mathbf{z}_s} \log p_\theta(\mathbf{z}_s|\mathbf{z}_t)$$

Introduces an extra term dependent on a property score y that modifies the unconditional logits

Classifier-Free Guidance

$$\nabla_{\mathbf{z}_s} \log p^{\gamma}(\mathbf{z}_s|y, \mathbf{z}_t) = \gamma \nabla_{\mathbf{z}_s} \log p_\theta(\mathbf{z}_s|y, \mathbf{z}_t) + (1-\gamma) \nabla_{\mathbf{z}_s} \log p_\theta(\mathbf{z}_s|\mathbf{z}_t)$$

Challenges of Guidance in the Discrete State Space

How do we guide the discrete diffusion process where the gradient is undefined?

Classifier-Based Guidance

$$\nabla_{\mathbf{z}_s} \log p^{\gamma}(\mathbf{z}_s|y,\mathbf{z}_t) = \gamma \nabla_{\mathbf{z}_s} \log p(y|\mathbf{z}_s) + \nabla_{\mathbf{z}_s} \log p_\theta(\mathbf{z}_s|\mathbf{z}_t)$$

Introduces an extra term dependent on a property score y that modifies the unconditional logits

Classifier-Free Guidance

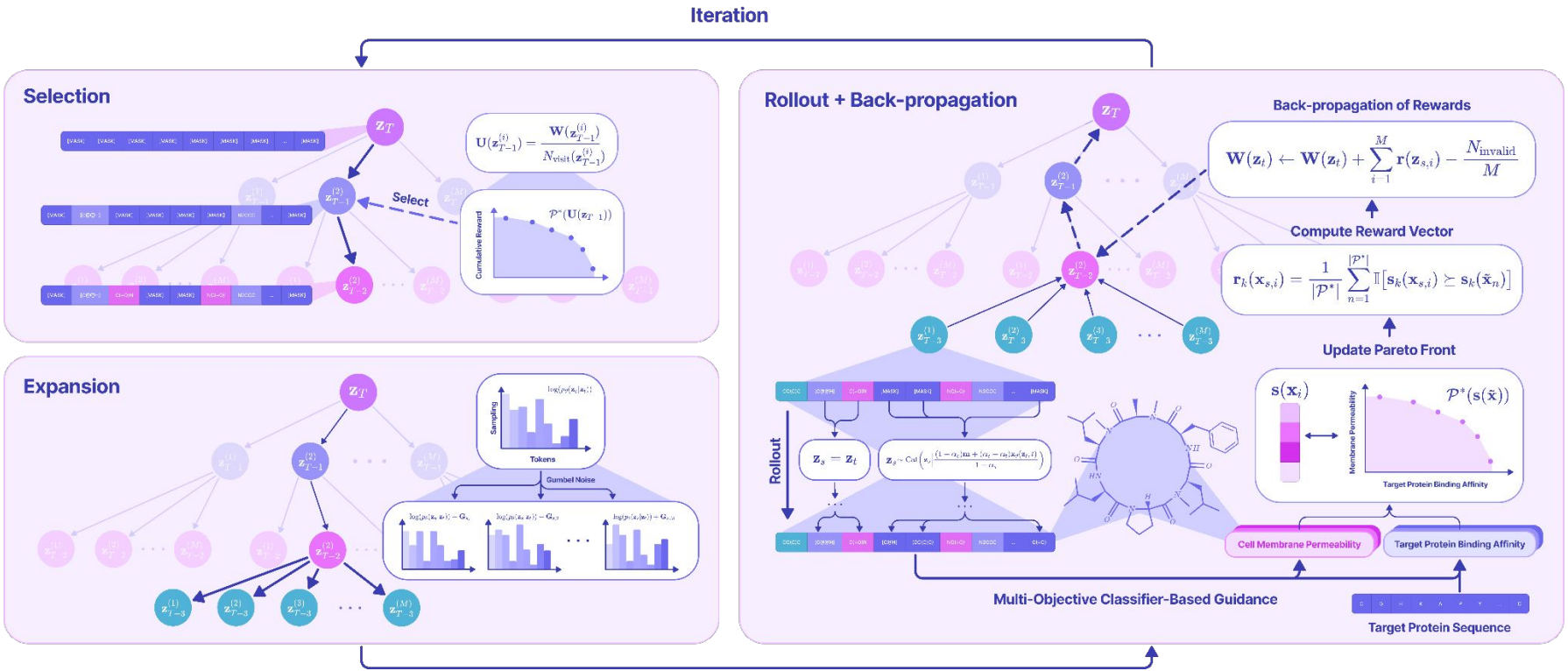
$$\nabla_{\mathbf{z}_s} \log p^{\gamma}(\mathbf{z}_s|y,\mathbf{z}_t) = \gamma \nabla_{\mathbf{z}_s} \log p_\theta(\mathbf{z}_s|y,\mathbf{z}_t) + (1-\gamma)\nabla_{\mathbf{z}_s} \log p_\theta(\mathbf{z}_s|\mathbf{z}_t)$$

Limitations

1. Not compatible with multi-objective guidance
2. Does not balance conflicting objectives
3. Requires evaluating rewards on intermediate, noisy sequences
4. Only produces a single sequence for each run through the model

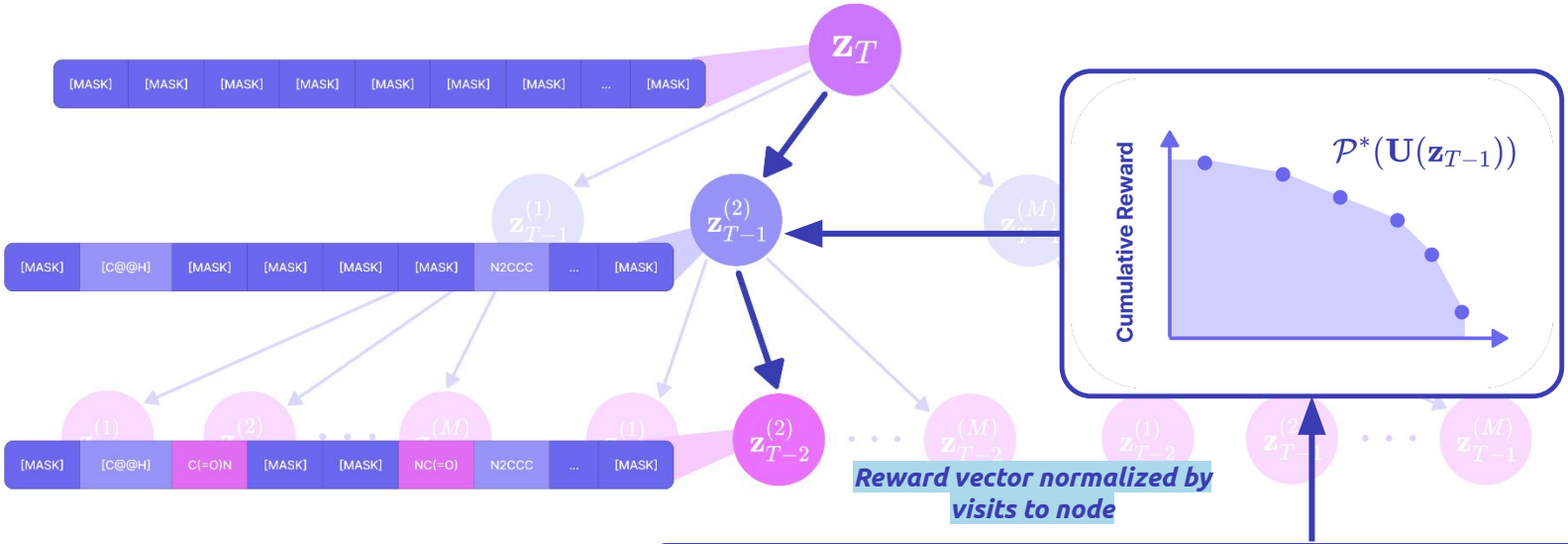
Monte Carlo Tree Guidance

Iteratively exploring unmasking paths to discover sequences optimized across multiple objectives



Selection of Unmasking Path with High Reward

Traversing the unmasking path with high potential reward



Reward vector normalized by visits to node

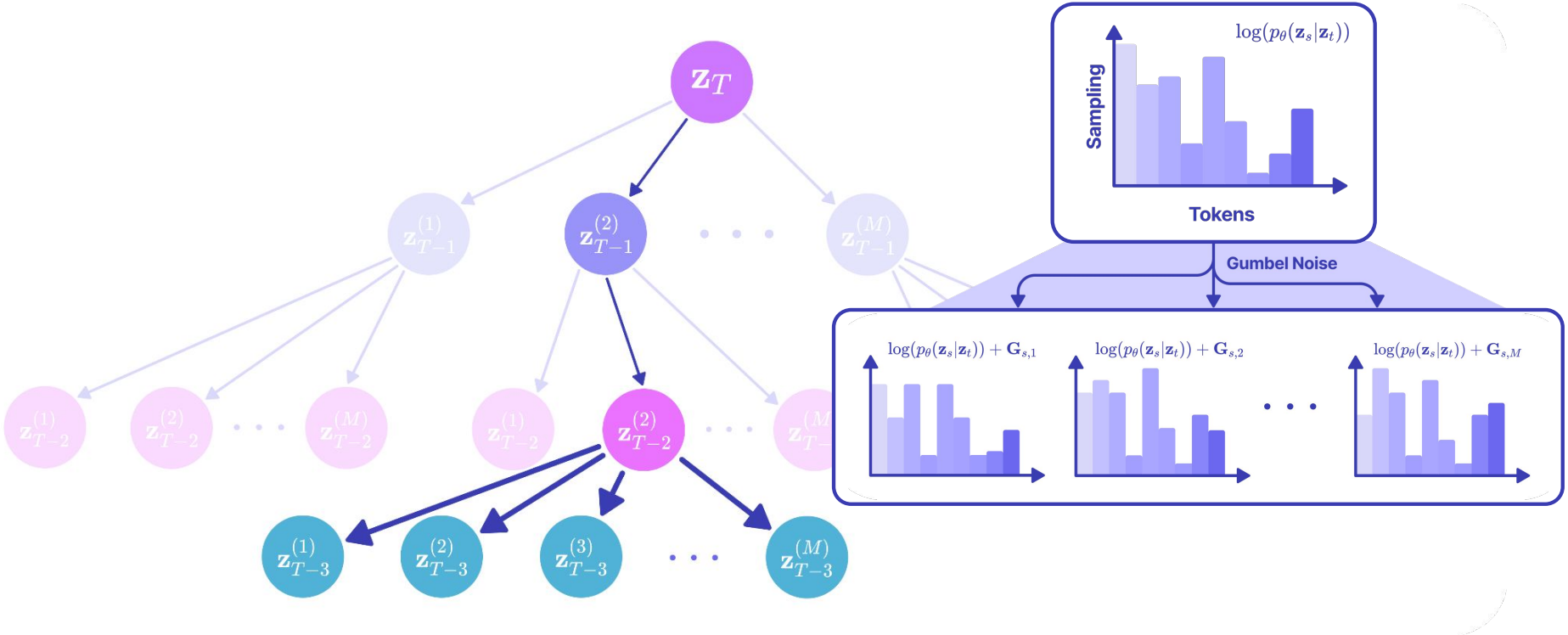
$$\mathbf{U}(\mathbf{z}_t, \mathbf{z}_{s,i}) = \frac{\mathbf{W}(\mathbf{z}_{s,i})}{N_{\text{visit}}(\mathbf{z}_{s,i})} + c \cdot p_\theta(\mathbf{z}_{s,i}|\mathbf{z}_t) \frac{\sqrt{N_{\text{visit}}(\mathbf{z}_t)}}{1 + N_{\text{visit}}(\mathbf{z}_{s,i})}$$

Selection score vector

Encourages exploration to unvisited child nodes

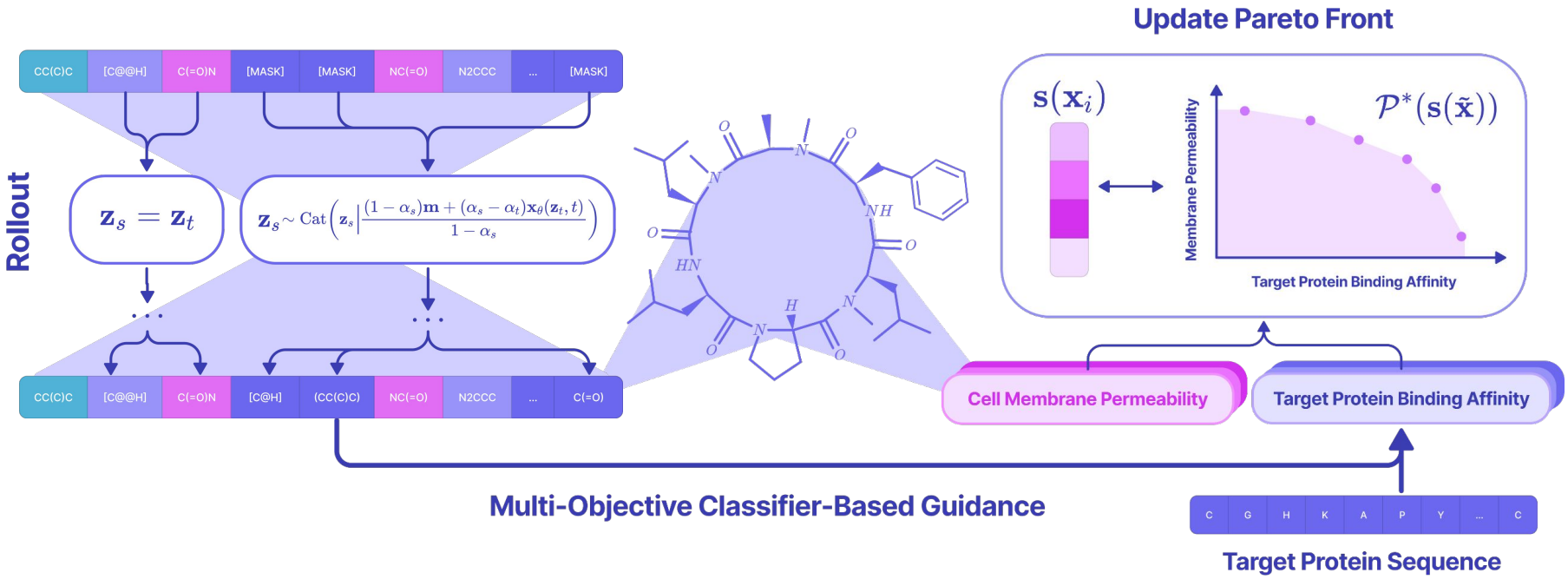
Expansion of Diverse Unmasking Strategies

Exploring multiple unmasking steps by applying stochastic Gumbel noise



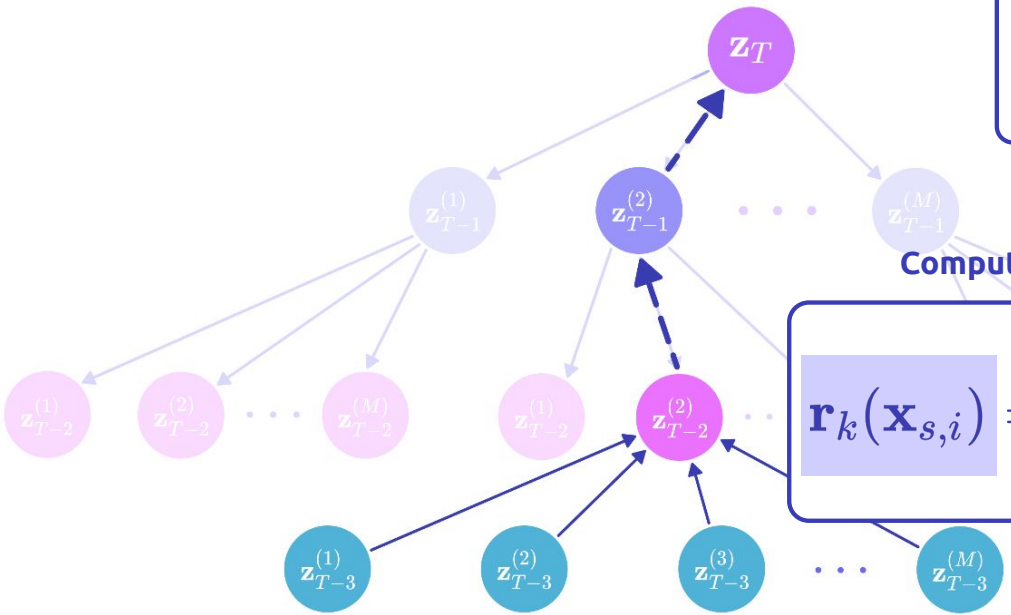
Rollout to Clean Sequence for Reward Evaluation

Obtaining a clean sequence from each expanded node with greedy argmax sampling



Backpropagation of Multi-Objective Reward Vector

Biasing the next iteration on paths that have high probability of generating high-reward sequences



Add Child Rewards to All Predecessor Nodes

$$\mathbf{W}(\mathbf{z}_t) \leftarrow \mathbf{W}(\mathbf{z}_t) + \sum_{i=1}^{M} \mathbf{r}(\mathbf{z}_{s,i})$$

Sum of child rewards

Compute Reward Vector for Each Rollout Sequence

$$\mathbf{r}_k(\mathbf{x}_{s,i}) = \frac{1}{|\mathcal{P}^*|} \sum_{n=1}^{|\mathcal{P}^*|} \mathbb{I}\left[\mathbf{s}_k(\mathbf{x}_{s,i}) \succeq \mathbf{s}_k(\tilde{\mathbf{x}}_n)\right]$$

Number of sequences in the Pareto set that the sequence dominates

Iterate to Obtain Pareto Optimal Set

A total of N iterations and M child sequences per iteration, the Pareto non-dominated sequences are returned

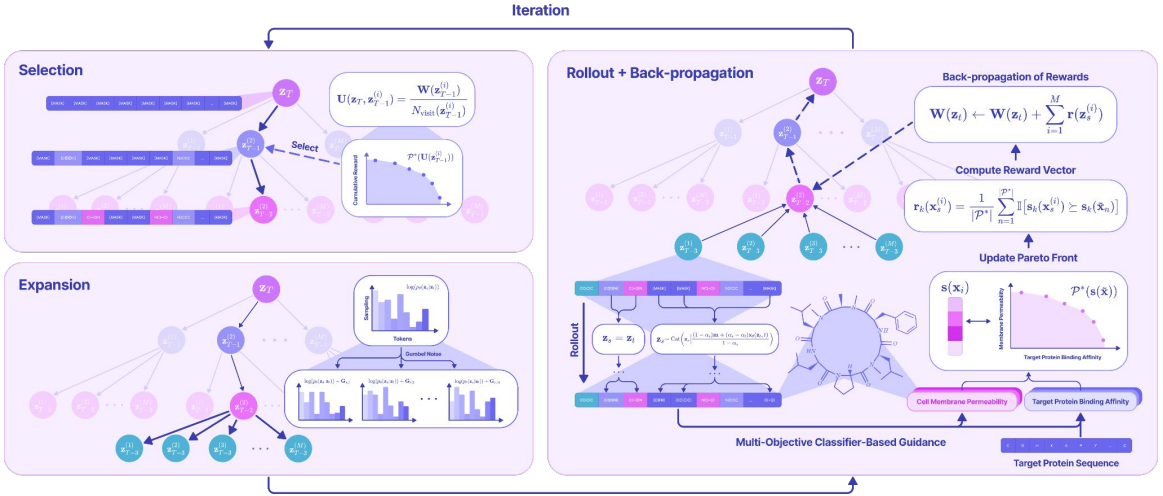
Selection

Start from a fully masked sequence (root node) and follow a sequence of *optimal* unmasking steps to a leaf node (unexpanded partially masked sequence)

Expansion

From the probability distribution generated from the trained diffusion model, apply Gumbel noise and sample *M* distinct partially unmasked sequences.

$$\log \tilde{p}_{\theta,i}(\mathbf{z}_{s,i}|\mathbf{z}_t) = \log p_\theta(\mathbf{z}_{s,i}|\mathbf{z}_t) + \mathbf{G}_i$$



Rollout

From each partially unmasked child sequence, use greedy unmasking to fully unmask the sequence for scoring. Feed the unmasked sequences into a set of *K* classifiers to determine Pareto optimality

Backpropagation

Calculate a reward vector of the fraction of the Pareto-optimal set that a sequence has a greater or equal score. Add the rewards across child nodes and add to the rewards of all predecessor nodes

$$\mathbf{r}_k(\mathbf{x}_{s,i}) = \frac{1}{|\mathcal{P}^*|} \sum_{n=1}^{|\mathcal{P}^*|} \mathbb{I}\left[\mathbf{s}_k(\mathbf{x}_{s,i}) \succeq \mathbf{s}_k(\tilde{\mathbf{x}}_n)\right]$$

General Framework for Multi-Objective Guidance for Discrete Diffusion

Our multi-objective guidance method can be applied to various modalities like mRNA, small molecules, DNA/RNA, and reasoning

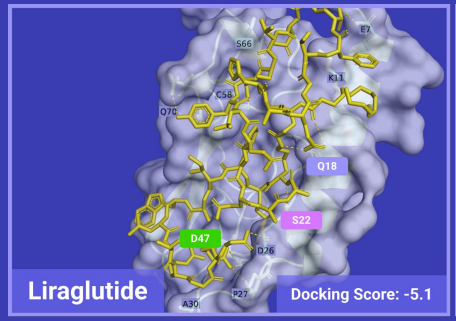
1. **Scales to *Any* Set of Properties Without Additional Training** (Except for the Classifier)
2. **Balances Exploration of Diverse Unmasking Paths and Exploitation of High-Reward Paths**
3. **Handles Conflicting Properties by Maintaining a Set of Pareto Non-Dominated Sequences**
4. **Can Optimize for Binding Affinity with Multiple Protein Target Sequences**

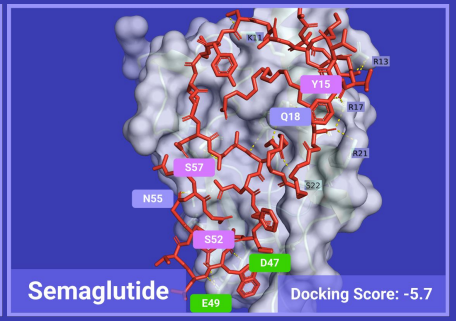
Multi-Objective Peptide SMILES Generation

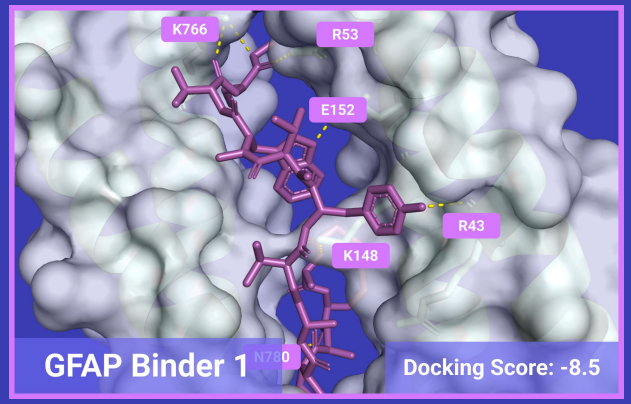
PepTune generates *valid* peptide SMILES optimized for binding affinity

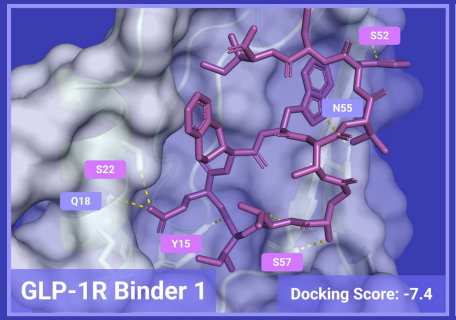
	Permeability Data	Binding Data	PepMDLM
Mean nAAs Per Peptide	2.215	2.150	2.940
Cyclic Peptides (%)	0.467	0.027	0.100

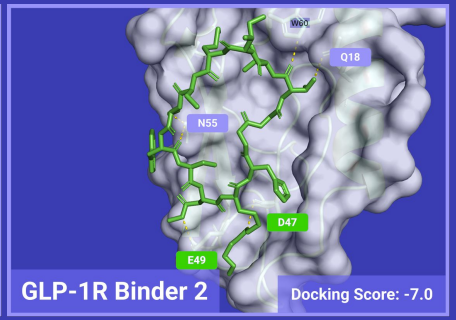
Model	Validity (↑)	Uniqueness (↑)	Diversity (↑)	SNN (↓)	Randomness (↑)	KL-Divergence (↑)
Data	1.000	1.000	0.885	1.000	4.55	0 (Reference)
PepMDLM	0.450	1.000	0.705	0.513	4.11	0.174
PepTune	1.000	1.000	0.677	0.486	4.12	0.173



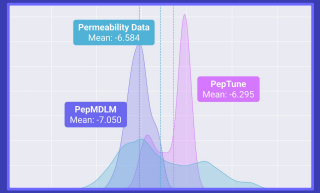




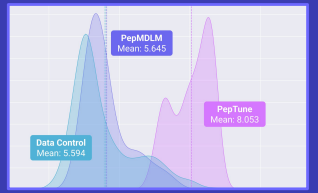




Membrane Permeability Density Plot



GFAP Binding Affinity Density Plot



Dual-Targeting Peptides to TfR and GLAST Protein

Peptides conditioned on affinity to TfR on the BBB and glutamate-aspartate transporter (GLAST) on astrocytes show enhanced docking scores to both targets

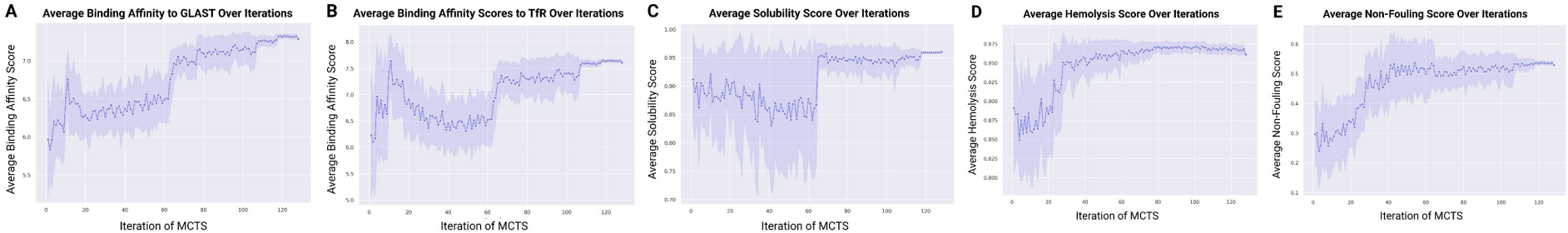


Table 6: PepTune-generated dual-target binders to TfR and GLAST.

Binder ID	**TfR Docking Score** (kcal/mol) (↓)	**GLAST Docking Score** (kcal/mol) (↓)	**Solubility** (↑)	**Hemolysis** (↑)	**Non-fouling** (↑)
Binder 1	-8.8 (8.800)	-8.9 (7.775)	**0.975**	0.743	0.118
Binder 2	-8.0 (7.599)	-7.9 (6.751)	0.938	0.835	**0.309**
Binder 3	-8.3 (7.537)	-8.2 (6.662)	0.972	**0.914**	0.214
Binder 4	-7.6 (7.748)	-7.5 (6.946)	0.959	0.902	0.290
Binder 5	**-10.5** (8.714)	-8.5 (7.398)	0.811	0.748	0.202
Binder 6	-8.4 (8.197)	-7.5 (7.076)	0.971	0.855	0.165
Binder 7	-9.3 (8.321)	**-9.2** (7.190)	0.881	0.860	0.212

[1] The predicted binding affinity scores by our trained classifier are placed in brackets beside the docking score. Larger scores indicate stronger binding for our classifier.

Conclusions

Why is PepTune significant?

1. **Modulating masking schedules** for different tokens can be useful for **preserving fundamental structures in data beyond just peptide bonds**
2. **Monte Carlo Tree Guidance** provide a generalized framework for multi-objective guidance for discrete diffusion models that **requires no additional training** and can **scale to any set of objectives**

MCTG enables effective multi-objective guidance of discrete diffusion *without* additional training, **but what if we want the model to *inherit* the multi-objective capabilities?**

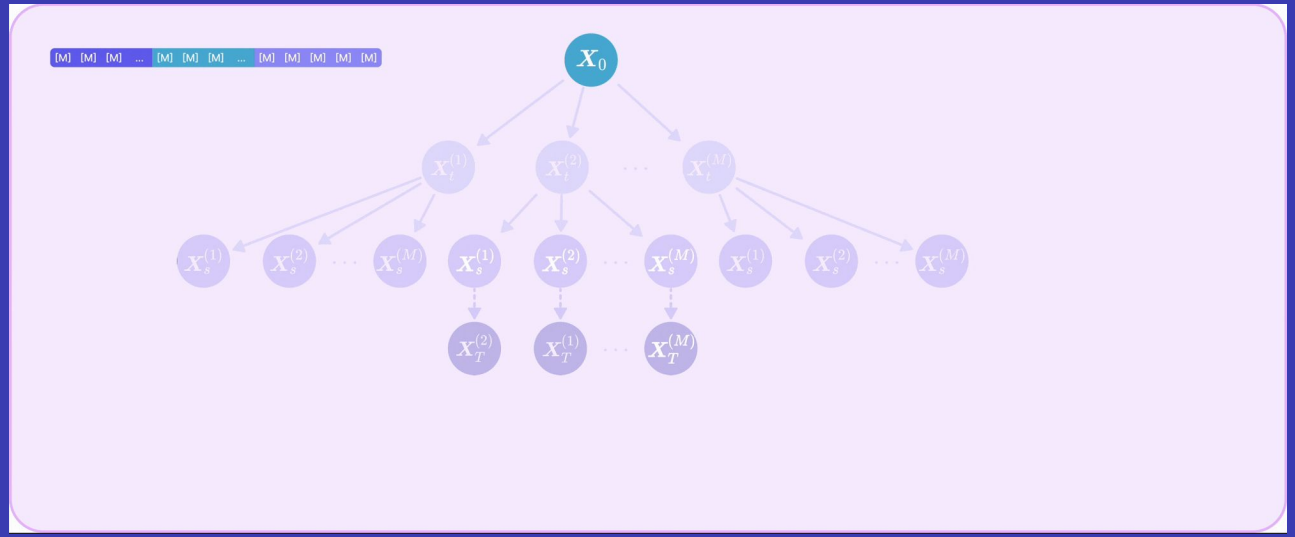
How can we amortize the search cost by improving the underlying model weights?

Off-policy RL enables learning from diffusion trajectories from a replay buffer that is reweighted to match the target reward-tilted distribution

Can we fine-tune using off-policy RL using diffusion trajectories optimized with MCTG?

TR2-D2: Tree Search Guided Trajectory-Aware Fine-Tuning for Discrete Diffusion

Sophia Tang* • **Yuchen Zhu*** • Molei Tao • Pranam Chatterjee



TR2-D2 Paper

Structured Search + Off-Policy RL = Discrete Diffusion Fine-Tuning

Generating optimal replay buffers for efficient and reliable off-policy RL fine-tuning

Algorithm 1 Framework for **Enhanced Reinforcement Learning with Structured Search**

Input: pre-trained model p^{pre}, finetune policy model p^{u_θ}, reward function r, and off-policy RL algorithm
Initalize finetune policy model $p^{u_\theta} = p^{\text{pre}}$
while not converged **do**
    while buffer $\mathcal{B}$ is not full **do**
        `Generate` samples from current policy model p^{u_θ}
        `Select` optimal samples that maximize the reward function and add to buffer
        `Explore` similar samples given previous selections using the `Search` algorithm
    end while
    `Update` θ using samples from $\mathcal{B}$ using off-policy RL algorithm for multiple epochs
    `Reset` the buffer $\mathcal{B}$
end while

TR2-D2 Outperforms PepTune in Multi-Objectives

Benchmark scores across multiple therapeutic properties of peptide SMILES

Target Protein	Method	Binding Affinity (↑)	Solubility (↑)	Non-hemolysis (↑)	Non-fouling (↑)	Permeability (↑)
TfR	Pre-trained	$8.008_{\pm 0.673}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	PepTune	$8.216_{\pm 0.703}$	$0.789_{\pm 0.144}$	$0.902_{\pm 0.051}$	$0.121_{\pm 0.081}$	$-7.389_{\pm 0.119}$
	TR2-D2 ($N_{\text{epochs}} = 200$)	$8.959_{\pm 0.796}$	$0.732_{\pm 0.145}$	$\mathbf{0.904}_{\pm \mathbf{0.038}}$	$0.229_{\pm 0.094}$	$-7.300_{\pm 0.067}$
	TR2-D2 ($N_{\text{epochs}} = 1000$)	$\mathbf{10.098}_{\pm \mathbf{0.050}}$	$\mathbf{0.838}_{\pm \mathbf{0.066}}$	$0.896_{\pm 0.012}$	$\mathbf{0.271}_{\pm \mathbf{0.038}}$	$\mathbf{-7.168}_{\pm \mathbf{0.024}}$
GLP-1R	Pre-trained	$8.233_{\pm 0.367}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	PepTune	$8.403_{\pm 0.365}$	$0.774_{\pm 0.170}$	$\mathbf{0.907}_{\pm \mathbf{0.057}}$	$0.125_{\pm .082}$	$-7.388_{\pm 0.128}$
	TR2-D2 ($N_{\text{epochs}} = 200$)	$9.059_{\pm 0.329}$	$0.700_{\pm 0.084}$	$0.839_{\pm 0.037}$	$0.385_{\pm .095}$	$-7.288_{\pm 0.047}$
	TR2-D2 ($N_{\text{epochs}} = 1000$)	$\mathbf{9.426}_{\pm \mathbf{0.035}}$	$\mathbf{0.841}_{\pm \mathbf{0.043}}$	$0.849_{\pm 0.016}$	$\mathbf{0.499}_{\pm \mathbf{0.037}}$	$\mathbf{-7.263}_{\pm \mathbf{0.020}}$
GLAST	Pre-trained	$7.830_{\pm 0.420}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	PepTune	$8.400_{\pm 0.353}$	$0.815_{\pm 0.139}$	$0.937_{\pm 0.029}$	$0.137_{\pm 0.086}$	$-7.311_{\pm 0.106}$
	TR2-D2 ($N_{\text{epochs}} = 200$)	$8.842_{\pm 0.274}$	$0.822_{\pm 0.122}$	$0.906_{\pm 0.031}$	$0.268_{\pm 0.086}$	$-7.316_{\pm 0.048}$
	TR2-D2 ($N_{\text{epochs}} = 1000$)	$\mathbf{9.703}_{\pm 0.072}$	$\mathbf{0.884}_{\pm 0.038}$	$\mathbf{0.930}_{\pm 0.007}$	$\mathbf{0.364}_{\pm 0.083}$	$\mathbf{-7.238}_{\pm 0.020}$
GFAP	Pre-trained	$7.084_{\pm 0.594}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	PepTune	$7.256_{\pm 0.704}$	$0.807_{\pm 0.167}$	$\mathbf{0.907}_{\pm 0.053}$	$0.124_{\pm 0.088}$	$-7.374_{\pm 0.134}$
	TR2-D2 ($N_{\text{epochs}} = 200$)	$8.539_{\pm 0.463}$	$0.820_{\pm 0.166}$	$0.905_{\pm 0.020}$	$\mathbf{0.154}_{\pm 0.043}$	$-7.256_{\pm 0.071}$
	TR2-D2 ($N_{\text{epochs}} = 1000$)	$\mathbf{9.762}_{\pm \mathbf{0.123}}$	$\mathbf{0.910}_{\pm 0.032}$	$0.889_{\pm 0.010}$	$0.137_{\pm 0.011}$	$\mathbf{-7.196}_{\pm 0.030}$
AMHR2	Pre-trained	$7.958_{\pm 0.253}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	PepTune	$8.284_{\pm 0.186}$	$0.789_{\pm 0.144}$	$\mathbf{0.930}_{\pm 0.039}$	$0.156_{\pm 0.074}$	$-7.346_{\pm 0.102}$
	TR2-D2 ($N_{\text{epochs}} = 200$)	$8.532_{\pm 0.117}$	$0.710_{\pm 0.192}$	$0.917_{\pm 0.047}$	$0.534_{\pm 0.144}$	$\mathbf{-7.159}_{\pm 0.073}$
	TR2-D2 ($N_{\text{epochs}} = 1000$)	$\mathbf{8.595}_{\pm 0.029}$	$\mathbf{0.947}_{\pm 0.0145}$	$0.923_{\pm 0.008}$	$\mathbf{0.766}_{\pm 0.023}$	$-7.164_{\pm 0.031}$
NCAM1	Pre-trained	$6.438_{\pm 0.372}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$\mathbf{0.102}_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	PepTune	$6.916_{\pm 0.240}$	$0.877_{\pm 0.105}$	$0.935_{\pm 0.039}$	$0.090_{\pm 0.075}$	$-7.391_{\pm 0.133}$
	TR2-D2 ($N_{\text{epochs}} = 200$)	$7.333_{\pm 0.186}$	$0.940_{\pm 0.065}$	$0.932_{\pm 0.047}$	$0.086_{\pm 0.117}$	$-7.123_{\pm 0.088}$
	TR2-D2 ($N_{\text{epochs}} = 1000$)	$\mathbf{7.541}_{\pm 0.025}$	$\mathbf{0.972}_{\pm 0.018}$	$\mathbf{0.974}_{\pm 0.003}$	$0.067_{\pm 0.009}$	$\mathbf{-6.930}_{\pm 0.028}$

Penn

Thanks for Listening!

Feel free to ask any questions or contact me at sophtang@seas.upenn.edu

PepTune Paper

TR2-D2 Paper