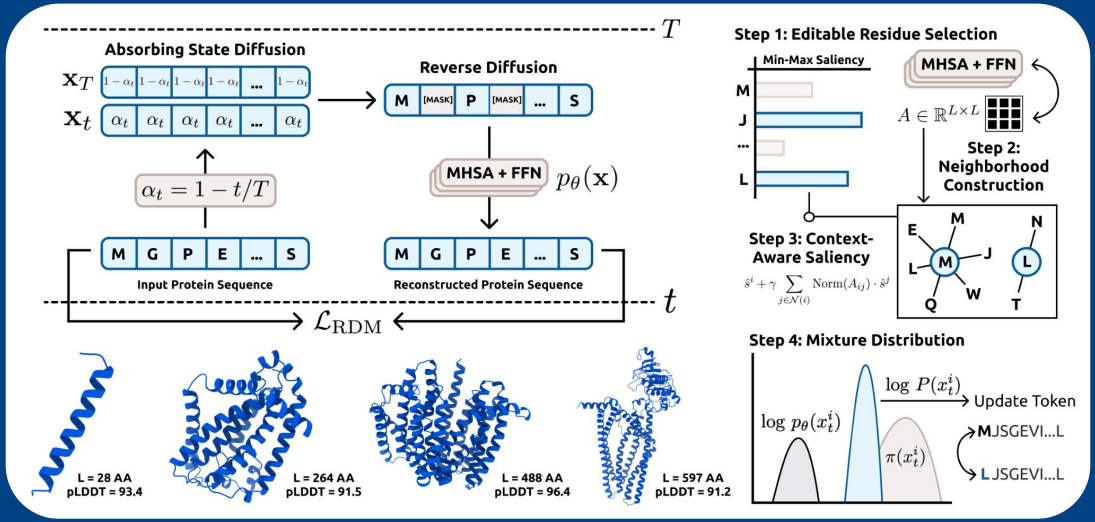
Duke | Penn

Token-Level Guided Discrete Diffusion for Membrane Protein Design

Shrey Goel • Perin Schray • Yinuo Zhang • Sophie Vincoff • Pranam Chatterjee



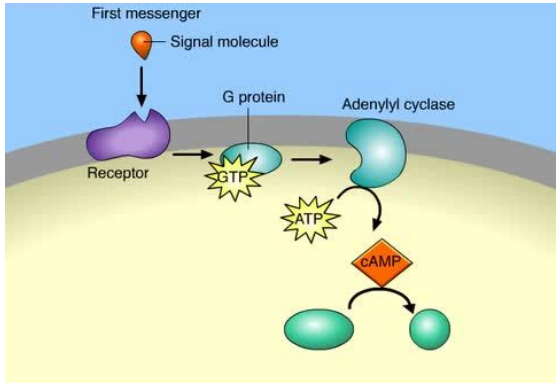
NeurIPS AI4Science Workshop | Dec 7, 2025

Significance of Membrane Proteins

Significance of Membrane Proteins

Signal Transduction

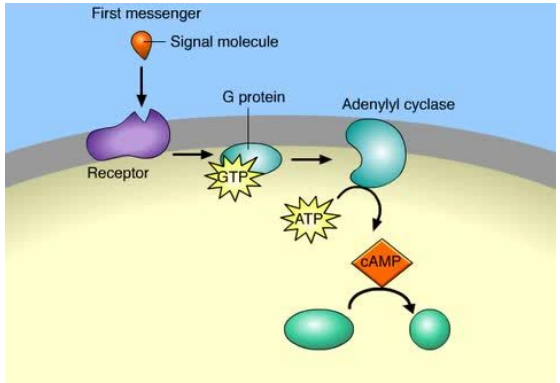
GPCRs, phosphorylation cascade



Significance of Membrane Proteins

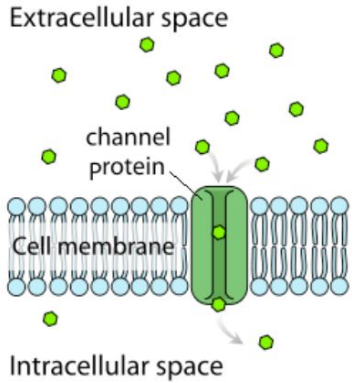
Signal Transduction

GPCRs, phosphorylation cascade



Molecular Transport

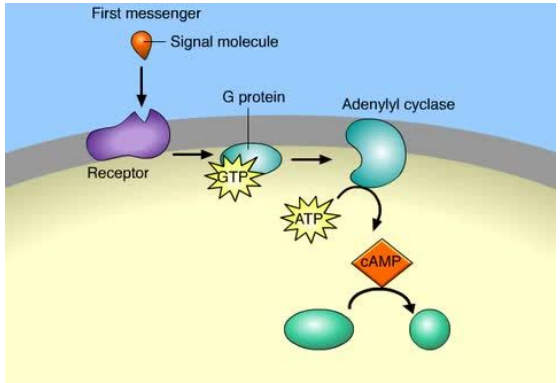
Nutrients, waste, maintain homeostasis



Significance of Membrane Proteins

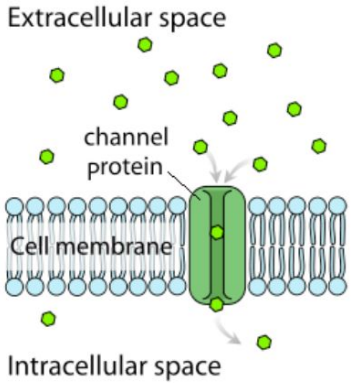
Signal Transduction

GPCRs, phosphorylation cascade



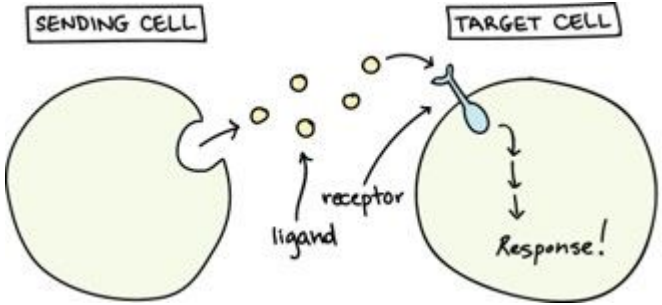
Molecular Transport

Nutrients, waste, maintain homeostasis

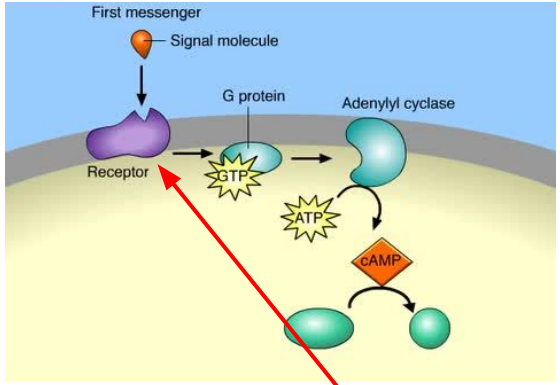


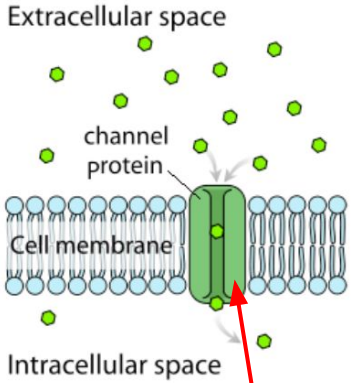
Cell Communication

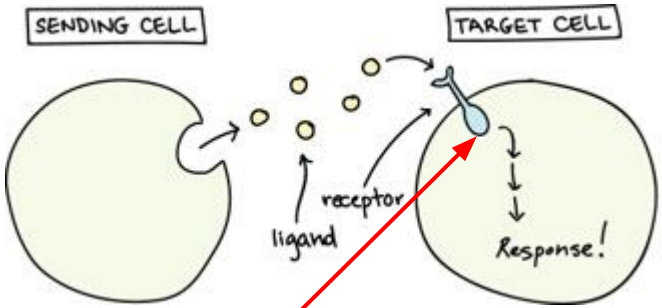
Gap junctions, receptors, cytokines



Significance of Membrane Proteins

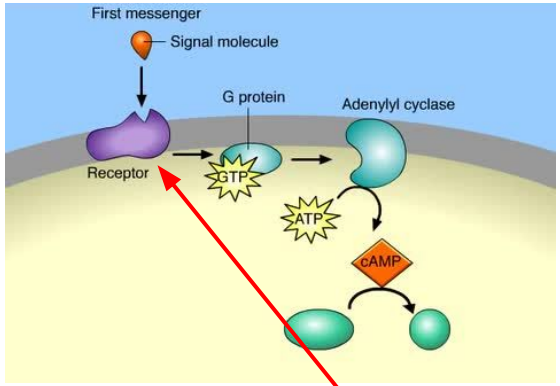


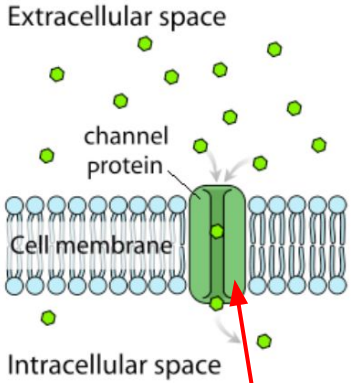


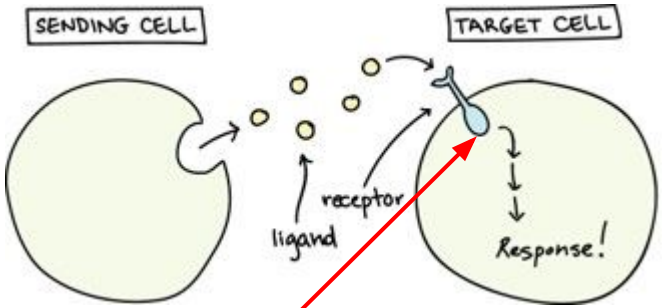


These are membrane proteins!

Significance of Membrane Proteins







These are membrane proteins!

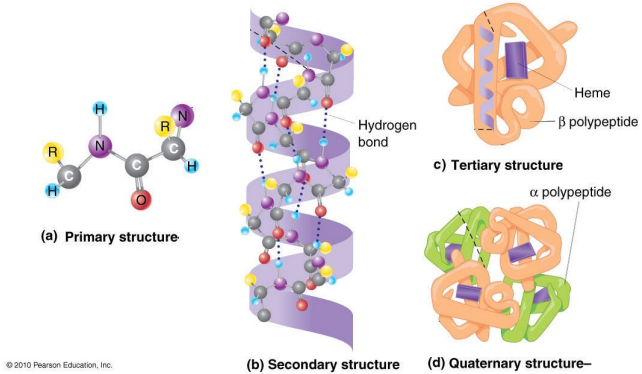
Design of membrane proteins is important!

Recall...

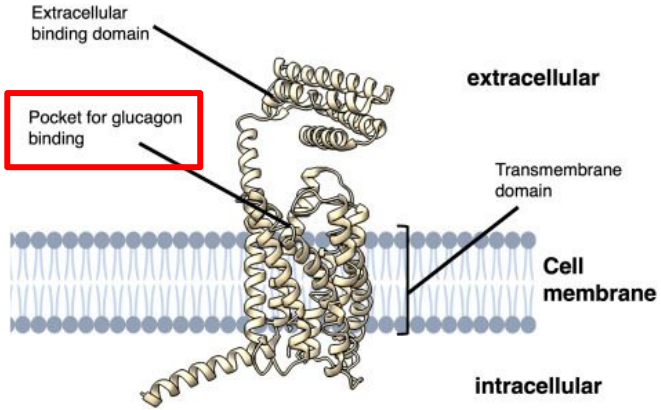
Sequence

RALNYRWQGWRYF...

Structure



Function

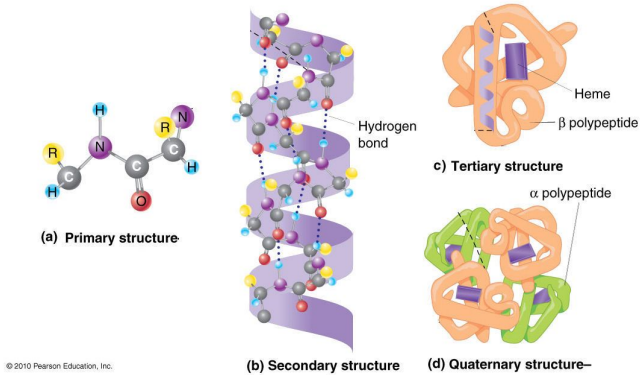


Recall…

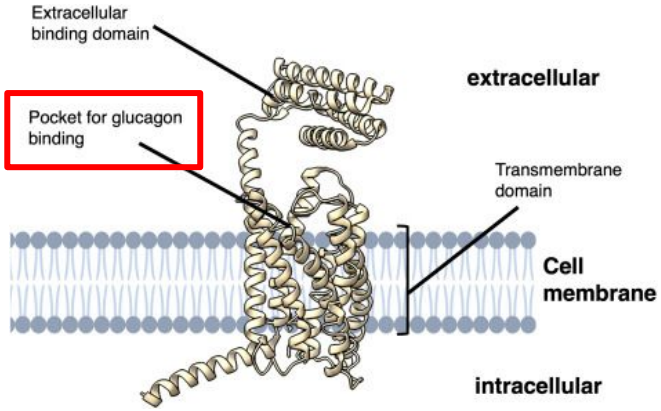
Sequence

RALNYRWQGWRYF…

Structure



Function



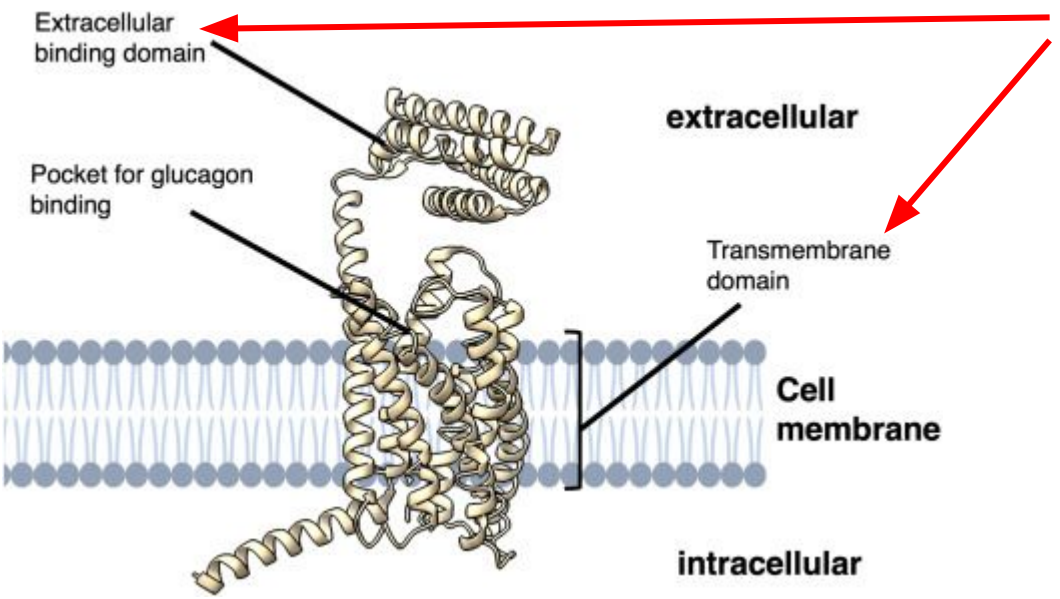
Current methods learn from structure!

Structure-based Membrane Protein Design

Problem: lack of experimentally solved structures

Structure-based Membrane Protein Design

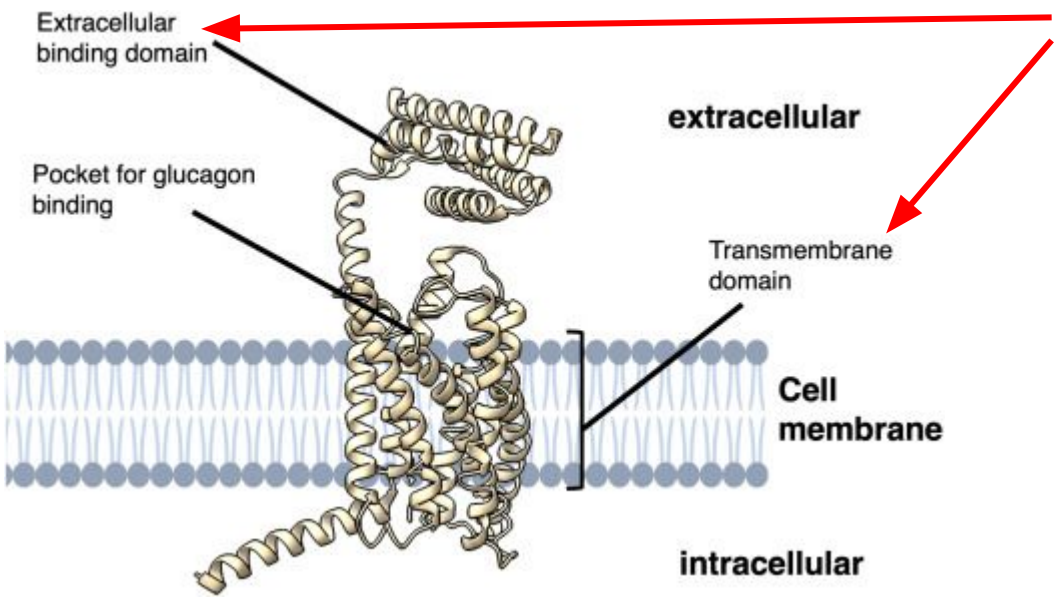
Problem: lack of experimentally solved structures



2 very different environments!

Structure-based Membrane Protein Design

Problem: lack of experimentally solved structures



2 very different environments!

"...structural characterization lags far behind that of globular proteins due to their naturally dual environment"

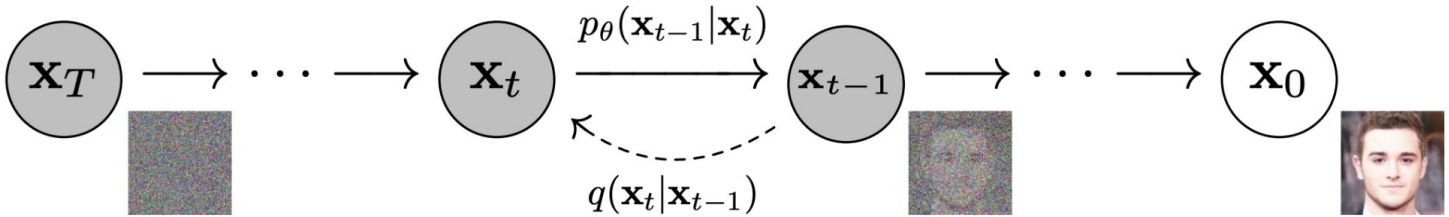
Jambrich…, Dobson. *(Scientific Reports)*

Diffusion Language Modeling

DDPMs

DDPMs

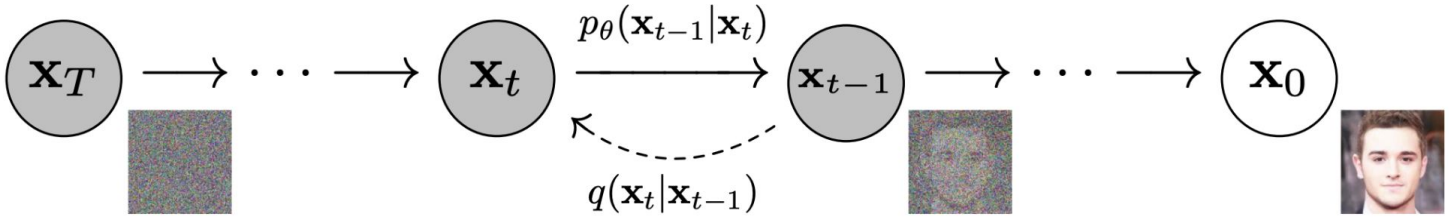
Training: learn the “denoising” process



Ho,…, Abbeel. *(arXiv)*

DDPMs

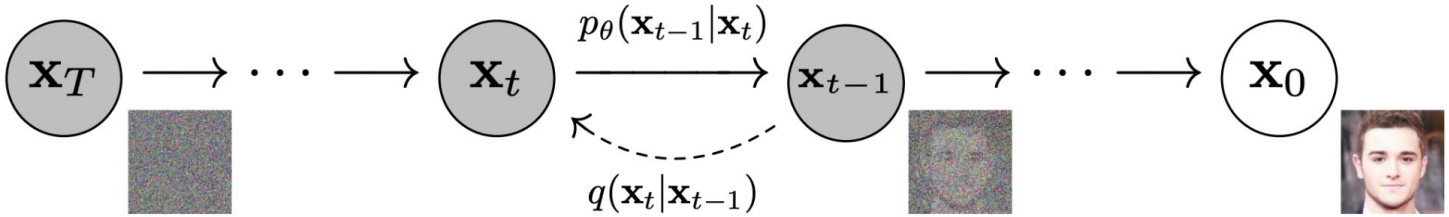
Generation: pure noise → clean sample



Ho,…, Abbeel. *(arXiv)*

Discrete Diffusion

Operate over categorical data (sequence tokens)

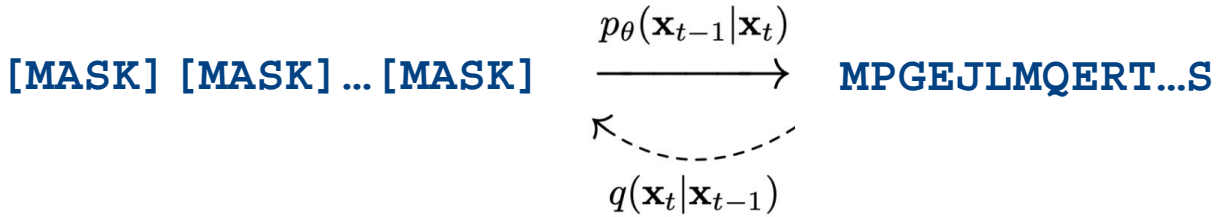


[MASK] [MASK] [MASK] … [MASK] $\xrightarrow{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}$ MPGEJLMQERT…S

$q(\mathbf{x}_t|\mathbf{x}_{t-1})$

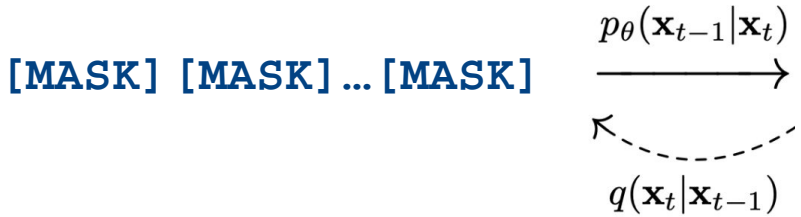
Discrete Diffusion

Forward: introduce discrete noise to categorical data (sequence tokens)



Discrete Diffusion

Forward: introduce discrete noise to categorical data (sequence tokens)

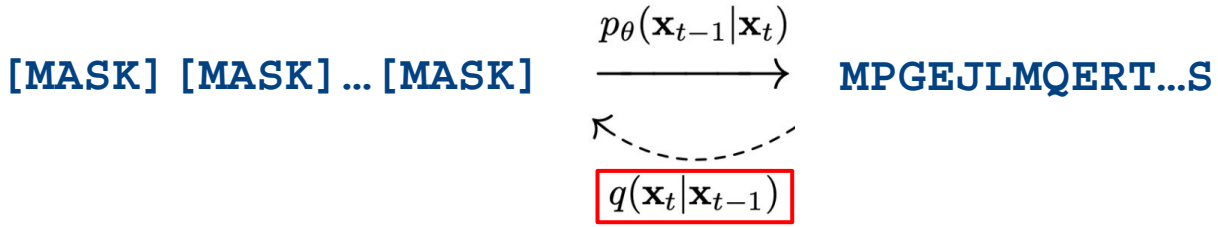


$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \alpha_t \, \mathbf{x}_0 + (1 - \alpha_t) \, \mathbf{q}_{\text{noise}}$$

$$\lim_{t \to T} \alpha_t = \lim_{t \to T} \left(1 - \frac{t}{T}\right) = 0$$

Discrete Diffusion

Forward: introduce discrete noise to categorical data (sequence tokens)

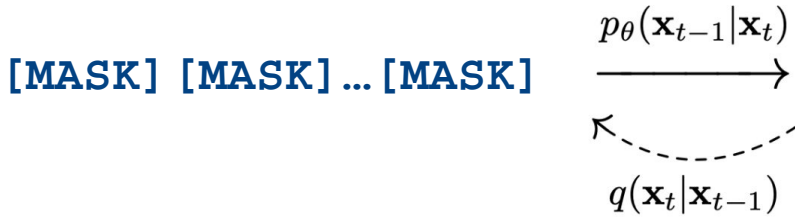


$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \alpha_t \, \mathbf{x}_0 + (1 - \alpha_t) \, \mathbf{q}_{\text{noise}}$$

$$\lim_{t \to T} \alpha_t = \lim_{t \to T} \left(1 - \frac{t}{T}\right) = 0$$

Discrete Diffusion

Forward: introduce discrete noise to categorical data (sequence tokens)



$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \alpha_t \, \mathbf{x}_0 + (1 - \alpha_t) \, \mathbf{q}_{\text{noise}}$$

$$\lim_{t \to T} \alpha_t = \lim_{t \to T} \left(1 - \frac{t}{T}\right) = 0$$

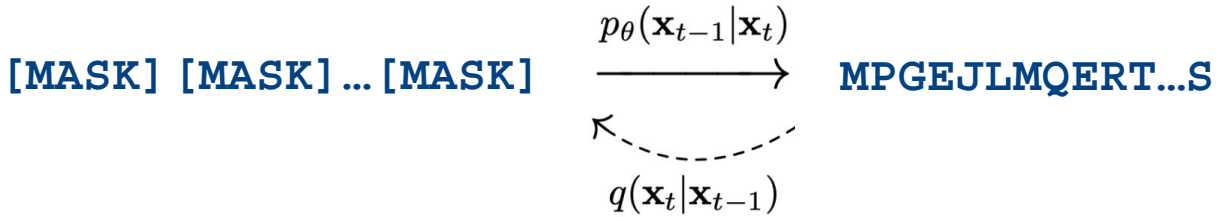
$$\Downarrow$$

$$x_t^i = \begin{cases} [\text{MASK}], & \text{if } u_i < \frac{t}{T}, \quad u_i \sim \text{Uniform}(0, 1) \\ x_0^i, & \text{otherwise} \end{cases}$$

$$\mathbf{x}_T = \{[\text{MASK}]\}_{i=1}^{L}$$

Discrete Diffusion

Forward: introduce discrete noise to categorical data (sequence tokens)



$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \alpha_t \, \mathbf{x}_0 + (1 - \alpha_t) \, \mathbf{q}_{\text{noise}}$$

$$\lim_{t \to T} \alpha_t = \lim_{t \to T} \left(1 - \frac{t}{T}\right) = 0$$

Larger timesteps → rely on masked prior
Smaller timesteps → rely on clean sequence

Discrete Diffusion

Training: learn the weighted log-likelihood of the data over masked tokens

$$\mathcal{L}_{\text{RDM}} = \mathbb{E}_{q(\mathbf{x}_0)} \left[\lambda_t \sum_{i=1}^{L} \mathbf{1}_{x_t^i \neq x_0^i} \cdot \log p_\theta(x_0^i \mid \mathbf{x}_t) \right]$$

Discrete Diffusion

Training: learn the weighted log-likelihood of the data over masked tokens

$$\mathcal{L}_{\text{RDM}} = \mathbb{E}_{q(\mathbf{x}_0)} \left[\lambda_t \sum_{i=1}^{L} \mathbf{1}_{x_t^i \neq x_0^i} \cdot \log p_\theta(x_0^i \mid \mathbf{x}_t) \right]$$

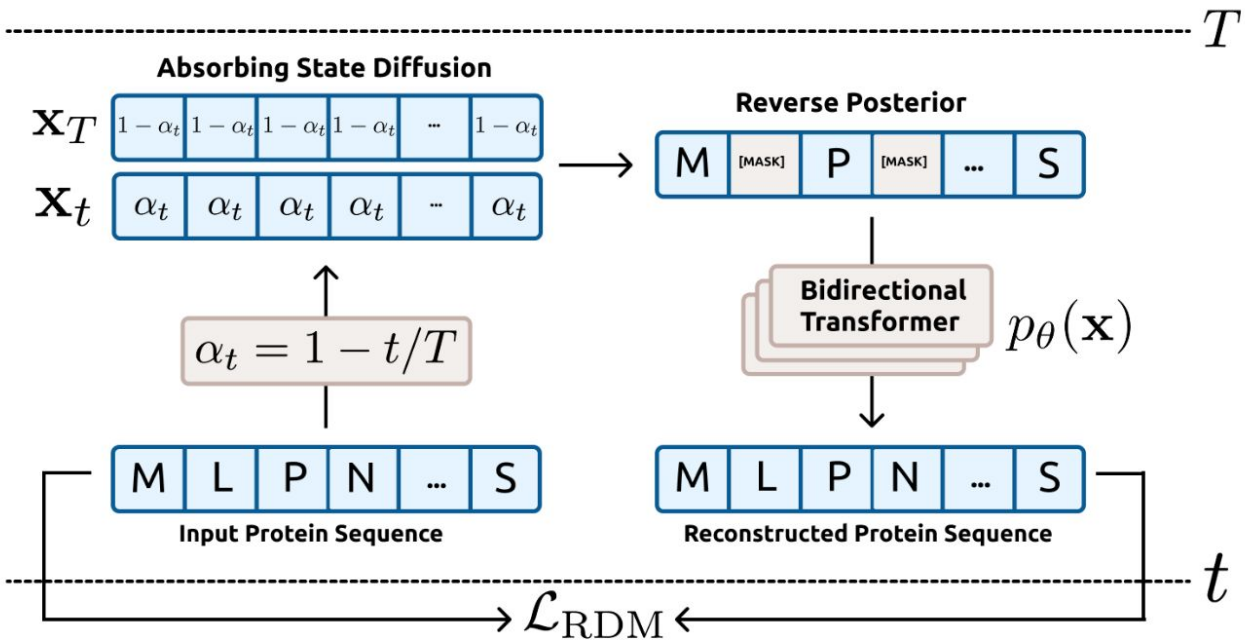
Larger timesteps → smaller weight

$$\lambda_t := T - (t-1)$$

Noisy sequences → harder training task

Zheng,..., Kong (arXiv)

MemDLM is a discrete diffusion language model that *de novo* generates membrane protein sequences

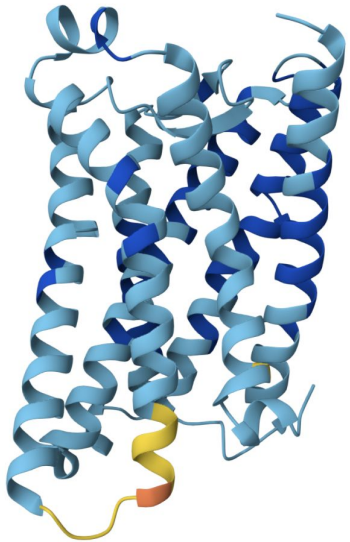


Classifier-Guided Sampling

Classifier-guided diffusion

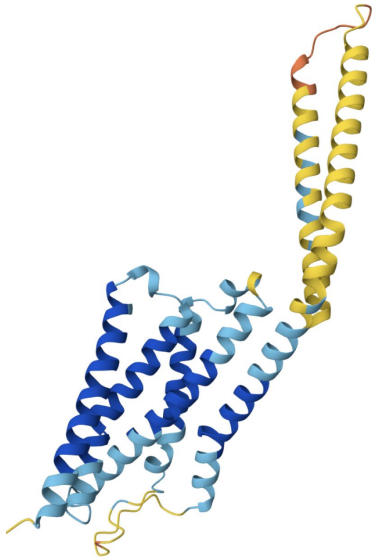
Can we control the generation?

De novo generated



How?

Solubilized



Classifier-guided diffusion

Can we control the generation?

$$\nabla_{\mathbf{x}_{t-1}} \log p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \nabla_{\mathbf{x}_{t-1}} \log p_\phi(y|\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Classifier-guided diffusion

Can we control the generation?

$$\nabla_{\mathbf{x}_{t-1}} \log p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \nabla_{\mathbf{x}_{t-1}} \log p_\phi(y|\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Classifier reward (solubility)

Classifier-guided diffusion

But wait 🛑

$$\nabla_{\mathbf{x}_{t-1}} \log p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \nabla_{\mathbf{x}_{t-1}} \log p_\phi(y|\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Classifier reward (solubility)

Classifier-guided diffusion

But wait 🛑

$$\nabla_{\mathbf{x}_{t-1}} \log p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \nabla_{\mathbf{x}_{t-1}} \log p_\phi(y|\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Classifier reward (solubility)

We must retain structurally-critical sequence tokens.

Classifier-guided diffusion

But wait 🛑

$$\nabla_{\mathbf{x}_{t-1}} \log p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \nabla_{\mathbf{x}_{t-1}} \log p_\phi(y|\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Classifier reward (solubility)

We must retain structurally-critical transmembrane residues.

Classifier-guided diffusion

But wait 🛑

RP**SWLASAL**ACG**NL**...Q**NFR**KEY

SLACADLIVGVLV...JNLYIVNN

Traditional classifier guidance algorithms do not guarantee the retention of targeted sequence tokens!

Per-Token Guidance (PET)

Components

1. Classifier model
2. Saliency map
3. Selecting editable residues
4. Neighborhood construction
5. Saliency update
6. Token update

Per-Token Guidance (PET)

Components

1. Classifier model
2. Saliency map
3. Selecting editable residues
4. Neighborhood construction
5. Saliency update
6. Token update

Per-Token Guidance (PET)

Classifier model

RPSWLASALACVLIFTIVVDILGNL...QNFRKEY

Per-Token Guidance (PET)

Classifier model

`RPSWLASALACVLIFTIVVDILGNL...QNFRKEY`

Solubility score for each residue

`0.23, 0.67, 0.10, 0.85, 0.92, …, 0.36`

Per-Token Guidance (PET)

Classifier model

`RPSWLASALACVLIFTIVVDILGNL...QNFRKEY`

Solubility score for each residue

`0.23, 0.67, 0.10, 0.85, 0.92, …, 0.36`

Identify uneditable residues

$$\mathcal{C} = \{i \in \{1, \ldots, L\} \mid v_\phi(h_t)_i \geq 0.5\}$$

Per-Token Guidance (PET)

Classifier model

`RPSWLASALACVLIFTIVVDILGNL...QNFRKEY`

Solubility score for each residue

`0.23, 0.67, 0.10, 0.85, 0.92, …, 0.36`

Identify uneditable residues

Uneditable = Already soluble!

$$\mathcal{C} = \{i \in \{1, \ldots, L\} \mid v_\phi(h_t)_i \geq 0.5\}$$

Per-Token Guidance (PET)

Components

1. Classifier model
2. Saliency map
3. Selecting editable residues
4. Neighborhood construction
5. Saliency update
6. Token update

Per-Token Guidance (PET)

Saliency map

- Saliency: how important is a token relative to the overall sequence solubility

Derived from Gruver, Stanton,... , Wilson (arXiv)

Per-Token Guidance (PET)

Saliency map

- Saliency: how important is a token relative to the overall sequence solubility

`RPSWLASALACVLIFTIVVDILGNL...QNFRKEY`

Saliency score for each residue

`0.22, 0.98, 0.02, 0.45, 0.92, …, 0.36`

Derived from Gruver, Stanton,... , Wilson (arXiv)

Per-Token Guidance (PET)

Saliency map

- Saliency: how important is a token relative to the overall sequence solubility

`RPSWLASALACVLIFTIVVDILGNL...QNFRKEY`

Saliency score for each residue

`0.22, 0.98, 0.02, 0.45, 0.92, …, 0.36`

$$\mathbf{s}(h) := \max\left\{\left(\sum_{d=1}^{D}|\nabla_h v_\phi(h)_d|\right)^{1/\tau}, \epsilon\right\}$$

Derived from Gruver, Stanton,... , Wilson (arXiv)

Per-Token Guidance (PET)

Saliency map

- Saliency: how important is a token relative to the overall sequence solubility

$$\mathbf{s}(h) := \max\left\{ \left(\sum_{d=1}^{D} |\nabla_h v_\phi(h)_d| \right)^{1/\tau}, \epsilon \right\}$$

High saliency score → token is important to the objective

Per-Token Guidance (PET)

Saliency map

- Saliency: how important is a token relative to the overall sequence solubility

$$\mathbf{s}(h) := \max\left\{\left(\sum_{d=1}^{D} |\nabla_h v_\phi(h)_d|\right)^{1/\tau}, \epsilon\right\}$$

High saliency score → token is already soluble

Per-Token Guidance (PET)

Remember from earlier?

$$\mathcal{C} = \{i \in \{1, \ldots, L\} \mid v_\phi(h_t)_i \geq 0.5\}$$

Per-Token Guidance (PET)

Remember from earlier?

$$\mathcal{C} = \{i \in \{1, \ldots, L\} \mid v_\phi(h_t)_i \geq 0.5\}$$

Augment the initial set of soluble residues with highly-salient tokens!

$$\mathcal{C} = \mathcal{C} \cup \text{top-}K\left(\hat{\mathbf{s}}, K = \max\left\{1,\ \tfrac{1}{10} \cdot (L - |\mathcal{C}|)\right\}\right)$$

Per-Token Guidance (PET)

What are we left with?

$$\mathcal{E} = \{1, \ldots, L\} \setminus \mathcal{C}$$

The complement represents our set of editable token indices
OR the tokens we can solubilize

Per-Token Guidance (PET)

Components

1. Classifier model
2. Saliency map
3. Selecting editable residues
4. Neighborhood construction
5. Saliency update
6. Token update

Per-Token Guidance (PET)

We can’t blindly modify the editable tokens

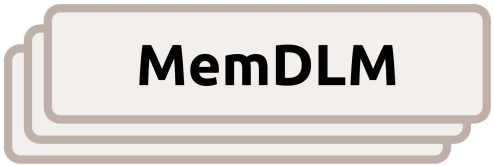
- How do we blend sequence-wide context into each token’s update?

M L

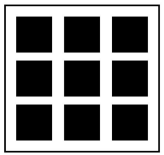
Per-Token Guidance (PET)

We can’t blindly modify the editable tokens

- How do we blend sequence-wide context into each token’s update?
- Attention scores give us this contextual information!



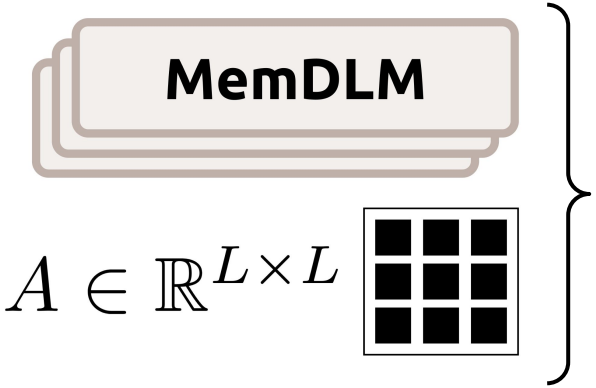
$$A \in \mathbb{R}^{L \times L}$$



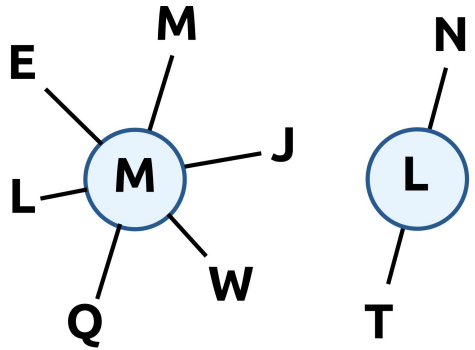
Per-Token Guidance (PET)

We can't blindly modify the editable tokens

- How do we blend sequence-wide context into each token's update?
- Attention scores give us this contextual information!



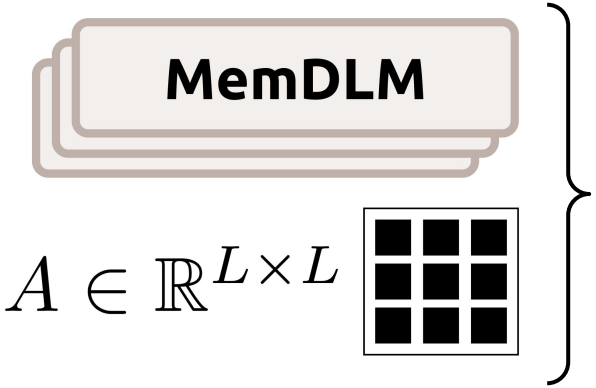
Create a *neighborhood* for each editable token



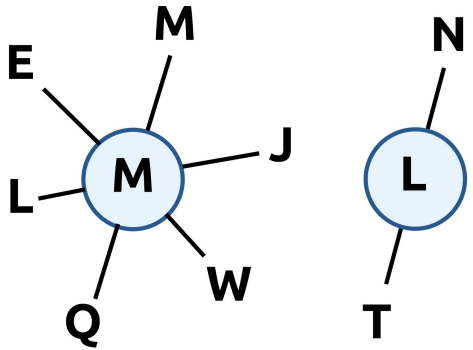
Per-Token Guidance (PET)

We can't blindly modify the editable tokens

- How do we blend sequence-wide context into each token's update?
- Attention scores give us this contextual information!



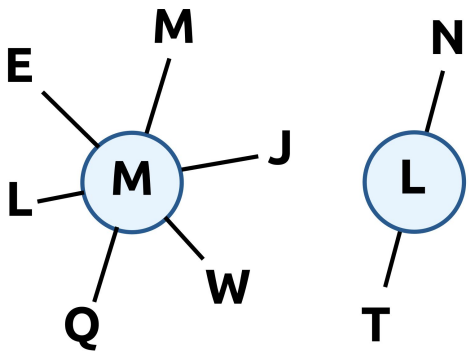
Create a *neighborhood* for each editable token based on the top-p attention scores



Per-Token Guidance (PET)

What do we have so far?

Neighborhoods for each editable token



RPSWLASALACVLIFTIVVDILGNL...QNFRKEY

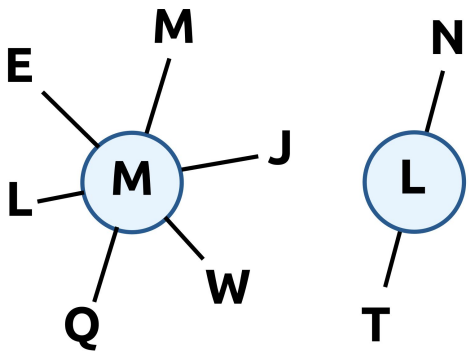
Saliency score for each residue

0.22, 0.98, 0.02, 0.45, 0.92, …, 0.36

Per-Token Guidance (PET)

What do we have so far?

Neighborhoods for each editable token



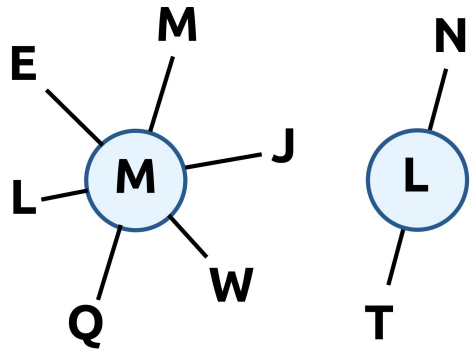
`RPSWLASALACVLIFTIVVDILGNL...QNFRKEY`

Saliency score for each residue

`0.22, 0.98, 0.02, 0.45, 0.92, …, 0.36`

We need to augment the saliency for each editable token with the saliency of the tokens in its neighborhood

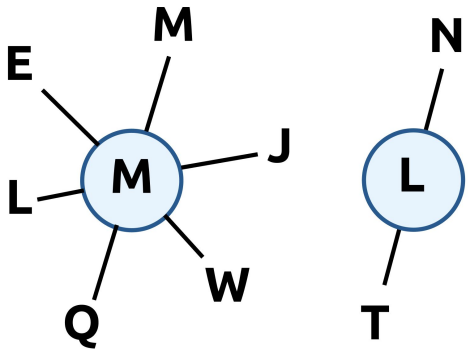
Per-Token Guidance (PET)



Saliency update

$$\tilde{s}^i := \hat{s}^i + \gamma \sum_{j \in \mathcal{N}(i)} \frac{A_{ij}}{\sum_{j' \in \mathcal{N}(i)} A_{ij'}} \cdot \hat{s}^j$$

Per-Token Guidance (PET)

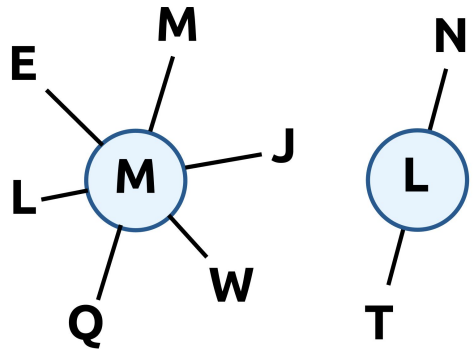


Saliency update

$$\tilde{s}^i := \hat{s}^i + \gamma \sum_{j \in \mathcal{N}(i)} \frac{A_{ij}}{\sum_{j' \in \mathcal{N}(i)} A_{ij'}} \cdot \hat{s}^j$$

Editable token's current saliency

Per-Token Guidance (PET)



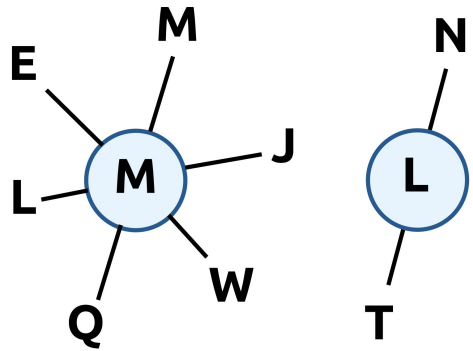
Saliency update

$$\tilde{s}^i := \hat{s}^i + \gamma \sum_{j \in \mathcal{N}(i)} \frac{A_{ij}}{\sum_{j' \in \mathcal{N}(i)} A_{ij'}} \cdot \hat{s}^j$$

Editable token's current saliency

Saliency score + attention score (importance) for neighborhood tokens

Per-Token Guidance (PET)



Saliency update

$$\tilde{s}^i := \hat{s}^i + \gamma \sum_{j \in \mathcal{N}(i)} \frac{A_{ij}}{\sum_{j' \in \mathcal{N}(i)} A_{ij'}} \cdot \hat{s}^j$$

Editable token's current saliency

Saliency score + attention score (importance) for neighborhood tokens

Now, each editable token's saliency includes sequence-wide context

Per-Token Guidance (PET)

Components

1. Classifier model
2. Saliency map
3. Selecting editable residues
4. Neighborhood construction
5. Saliency update
6. Token update

Per-Token Guidance (PET)

Token update

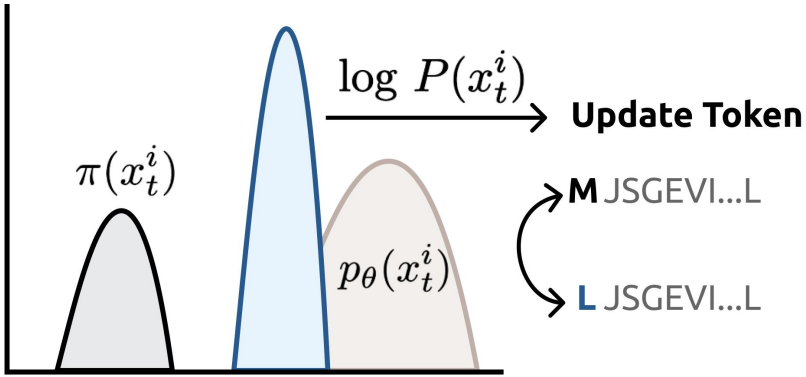
- Define weighting factor: $w_i = \sigma(\alpha \cdot \tilde{s}^i)$
- Prior distribution: $\pi(x_t^i) := \log p_\theta(x_{t-1}^i)$

Per-Token Guidance (PET)

Token update

- Define weighting factor: $w_i = \sigma(\alpha \cdot \tilde{s}^i)$
- Prior distribution: $\pi(x_t^i) := \log p_\theta(x_{t-1}^i)$
- Update step:

$$\log P(x_t^i) = (1 - w^i) \cdot \log p_\theta(x_t^i) + w^i \cdot \pi(x_t^i)$$

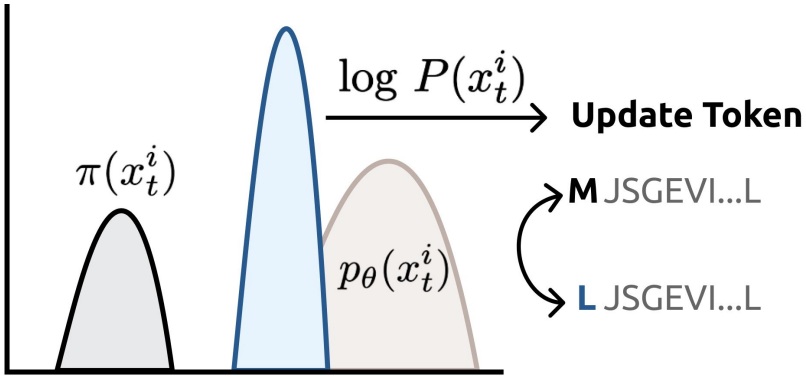


Per-Token Guidance (PET)

Token update

- Define weighting factor: $w_i = \sigma(\alpha \cdot \tilde{s}^i)$
- Prior distribution: $\pi(x_t^i) := \log p_\theta(x_{t-1}^i)$
- Update step:

$$\log P(x_t^i) = (1 - w^i) \cdot \log p_\theta(x_t^i) + w^i \cdot \pi(x_t^i)$$



High saliency → rely on original

Experiments

De Novo Generation

In silico benchmarks

	pLDDT (↑)	TMRD	PPL (↓)	Entropy (↑)
Test Set	76.637±10.676	0.294±0.219	5.707±3.435	3.918±0.253
ProGen2	54.998±19.235	0.048±0.153	126.646±1415.166	2.622±1.290
DPLM	62.318±20.669	0.310±0.264	**6.323**±10.317	3.179±0.812
MemDLM	**67.410**±14.828	**0.311**±0.250	6.344±3.278	**3.743**±0.326

TMRD: Transmembrane Residue Density

How membrane-like is the sequence?

De Novo Generation

In silico benchmarks

	pLDDT (↑)	TMRD	PPL (↓)	Entropy (↑)
Test Set	76.637±10.676	0.294±0.219	5.707±3.435	3.918±0.253
ProGen2	54.998±19.235	0.048±0.153	126.646±1415.166	2.622±1.290
DPLM	62.318±20.669	0.310±0.264	**6.323**±10.317	3.179±0.812
MemDLM	**67.410**±14.828	**0.311**±0.250	6.344±3.278	**3.743**±0.326

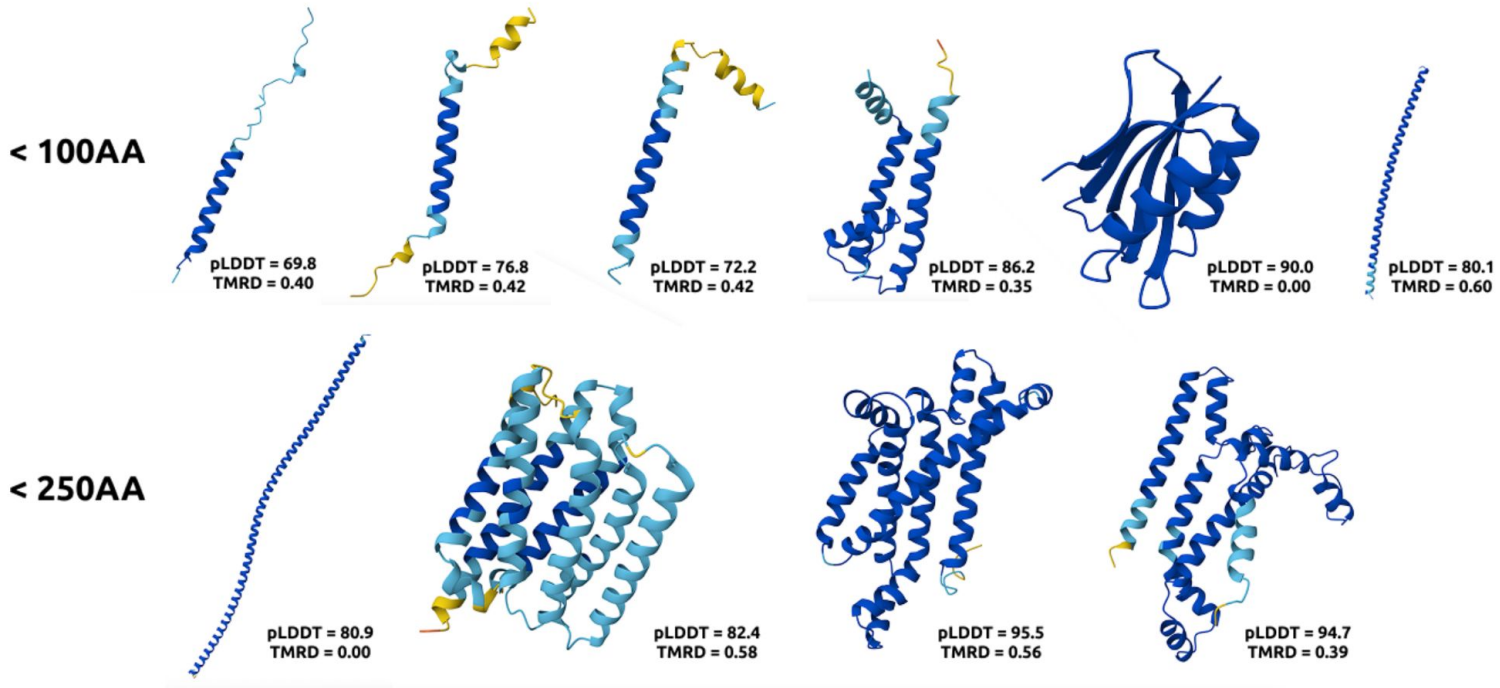
Key insights:

✓ High structural confidence (pLDDT)

✓ High token diversity (Shannon Entropy)

✓ High confidence (low perplexity)

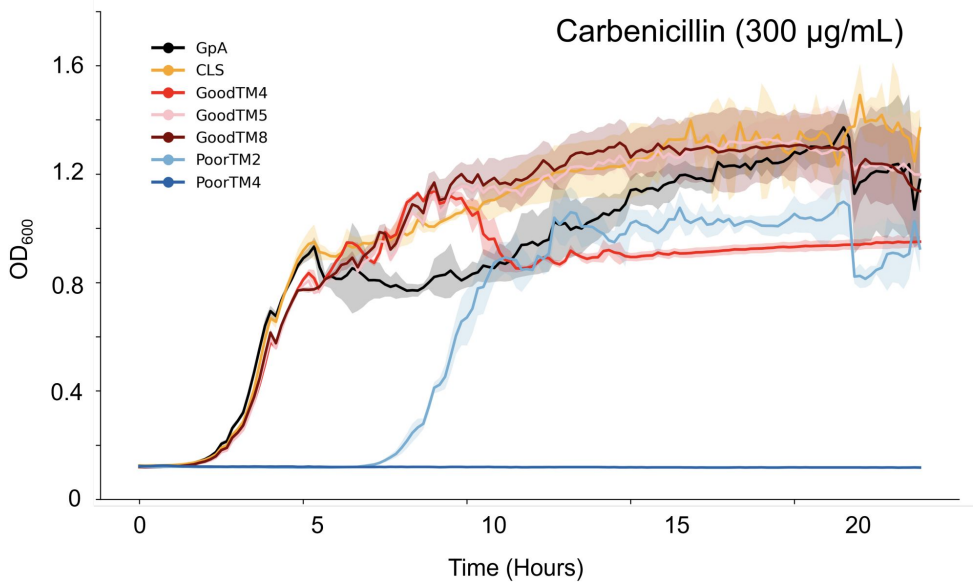
De Novo Generation



Key insights: MemDLM captures the underlying distrubtion of membrane protein sequences, producing single and multipass alpha-helical structures

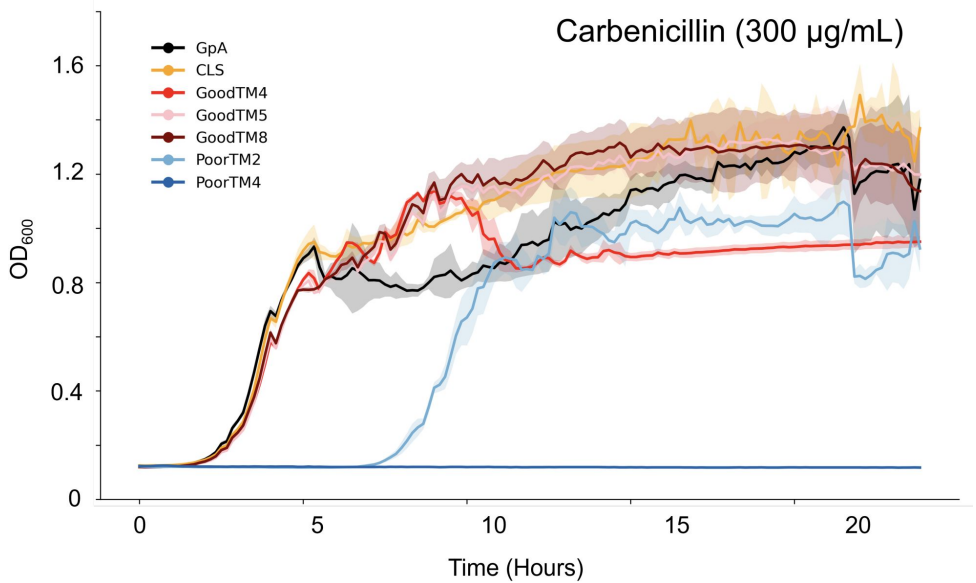
De Novo Generation

Experimental validation – TOXCAT β-Lactamase Assay



De Novo Generation

Experimental validation – TOXCAT β-Lactamase Assay



Key insight: MemDLM-generated sequences display growth kinetics similar to natural membrane protein sequences!

Classifier-guided diffusion

But wait 🛑

$$\nabla_{\mathbf{x}_{t-1}} \log p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \nabla_{\mathbf{x}_{t-1}} \log p_\phi(y|\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Classifier reward (solubility)

We must retain structurally-critical sequence tokens.

Solubilizing Membrane Proteins with PET

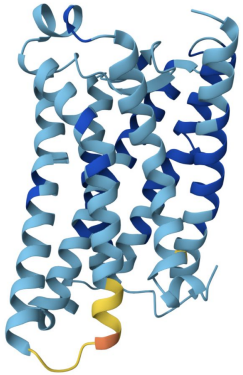
Use PET to solubilize existing membrane protein sequences

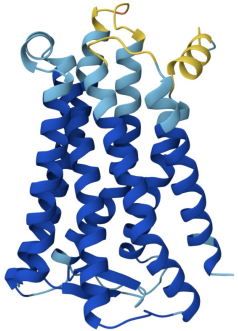
	pLDDT (↑)	TMRD (↓)	PPL (↓)	BLOSUM (↑)	Entropy (↑)
Test Set	$76.637_{\pm 10.676}$	$0.294_{\pm 0.219}$	$5.707_{\pm 3.435}$	–	$3.918_{\pm 0.253}$
MemDLM	$62.979_{\pm 17.906}$	$0.181_{\pm 0.192}$	$8.472_{\pm 4.879}$	$0.495_{\pm 2.346}$	$3.870_{\pm 0.268}$

Key insight: PET solubilizes membrane protein sequences while retaining structural confidence (BLOSUM score)

Solubilizing Membrane Proteins with PET

De novo generated

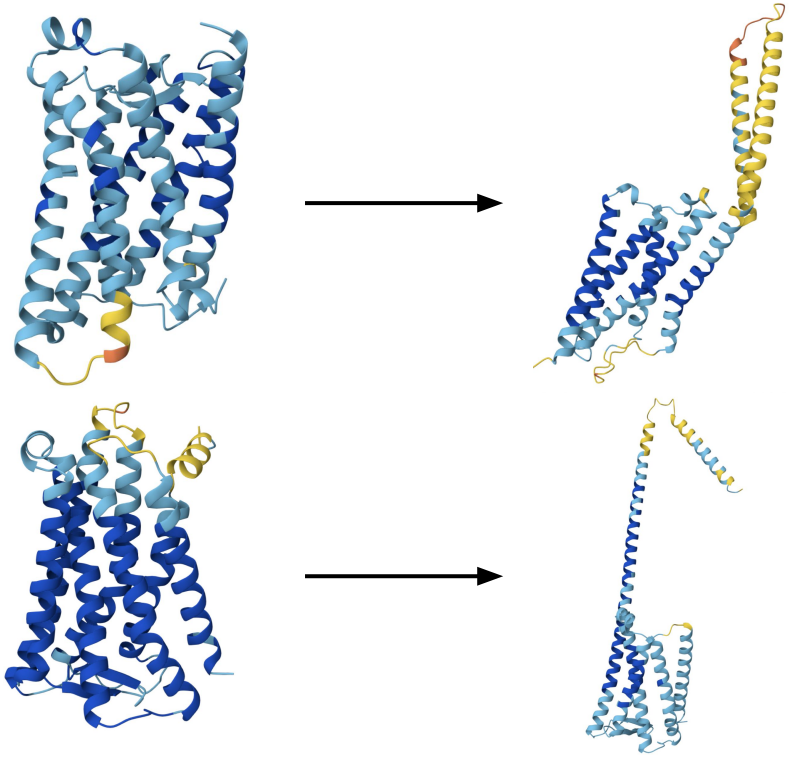




Solubilizing Membrane Proteins with PET

De novo generated

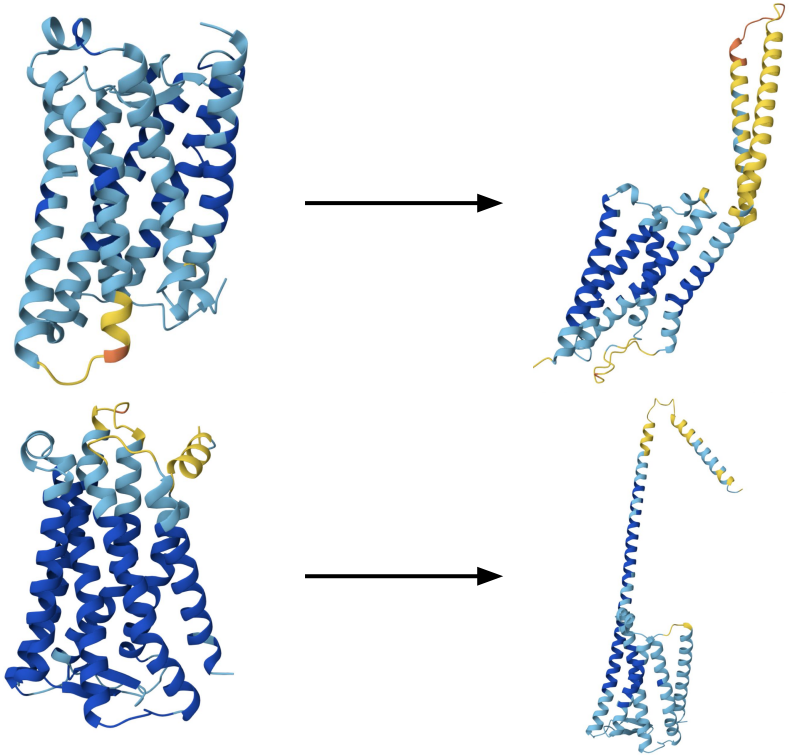
Solubilized



Solubilizing Membrane Proteins with PET

De novo generated

Solubilized



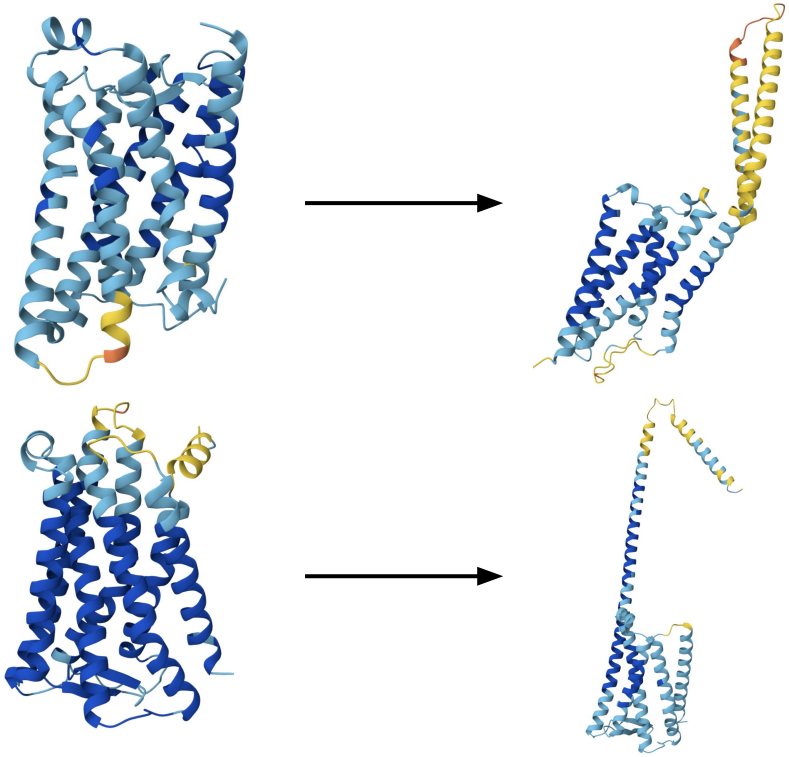
Key insights:

- PET solubilizes specific TM residues retaining an initial sequence scaffold
- Key extracellular domains AND structured alpha-helical domains are present

Solubilizing Membrane Proteins with PET

De novo generated

Solubilized



Key insights:

- PET solubilizes specific TM residues retaining an initial sequence scaffold
- Key extracellular domains AND structured alpha-helical domains are present

Next steps: Experimental test *de novo* soluble membrane proteins!

Duke UNIVERSITY

Penn
UNIVERSITY of PENNSYLVANIA

Thank you!

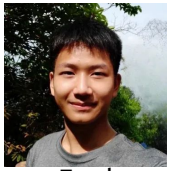
programmable.bio

Computational Team

(+ our other amazing computational members!)

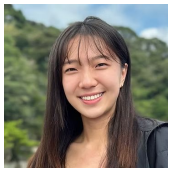

Yinuo


Sophie


Fred


Sophia


Tong


Rosie

Experimental Team

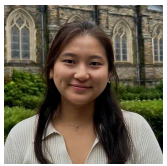
(+ our other amazing experimental members!)

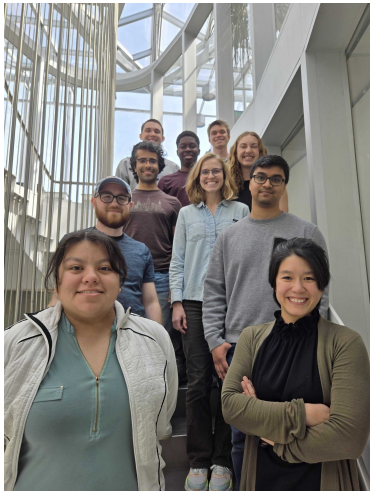

Lin


Zach


Tian


Divya


Lauren



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Perin

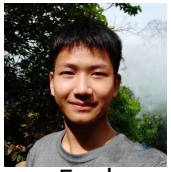
Thank you!

Computational Team

(+ our other amazing computational members!)

Yinuo

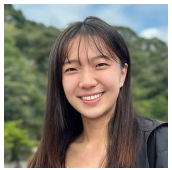
Sophie



Fred

Sophia

Tong



Rosie

Experimental Team

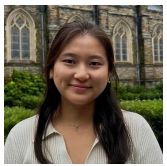
(+ our other amazing experimental members!)

Lin

Zach

Tian

Divya



Lauren

Duke UNIVERSITY

Penn
UNIVERSITY of PENNSYLVANIA

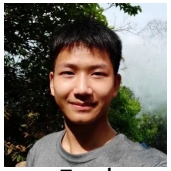
Thank you!

Computational Team

(+ our other amazing computational members!)

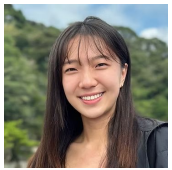

Yinuo


Sophie


Fred


Sophia


Tong


Rosie

Read the preprint!

Experimental Team

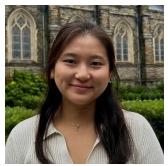
(+ our other amazing experimental members!)


Lin


Zach


Tian


Divya


Lauren

Motif Scaffolding

	Motif	pLDDT (↑)	PPL (↓)	BLOSUM (↑)	Entropy (↑)
Test Set	Insol	76.637±10.676	5.707±3.435	–	3.918±0.253
	Sol	76.637±10.676	5.707±3.435	–	3.918±0.253
EvoDiff	Insol	**64.058**±19.229	9.841±4.091	2.176±1.587	3.841±0.268
	Sol	64.036±19.145	4.632±3.271	-0.188±1.134	**3.841**±0.268
MemDLM	Insol	62.762±21.212	**8.748**±14.777	**2.964**±1.559	**3.876**±0.341
	Sol	**70.112**±16.912	**3.242**±2.362	**0.512**±1.556	3.803±0.321

Key insights:

- MemDLM scaffolds functional transmembrane and soluble domains, retaining structural confidence and biological plausibility
- Notably, infilled regions represent conserved substitutions

Representation Learning

Model	Solubility (↑)	Membrane Localization (↑)
ESM-2-650M	0.9383	0.6011
Fine-Tuned ESM-2	0.9375	0.6000
MemDLM	0.9375	0.5964

Key insights:

- MemDLM latent embeddings achieve predictive performance closely paralleling SOTA pLMs
- Even more performant than fine-tuning with a MLM objective!

Training Algorithm

Algorithm 1 MemDLM Training

Require: Protein sequence dataset $\mathcal{D}$, diffusion model p_θ, number of diffusion timesteps T

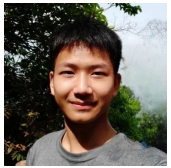
while not converged **do**
  Sample batch $\mathbf{x}_0 \sim \mathcal{D}$
  Sample timestep $t \sim \mathcal{U}(1, T)$
  Corrupt sequence: $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$
  Compute RDM loss: $\mathcal{L}_{\text{RDM}} = -\lambda_t \sum_{i=1}^{L} \log p_\theta(x_0^i \mid \mathbf{x}_t)$
  Take gradient descent step on: $\nabla_\theta \mathcal{L}_{\text{RDM}}$
end while
return Trained MemDLM p_θ

PET Sampling Algorithm

Algorithm 3 PET-based MemDLM Sampling

Require: Candidate protein sequence $\mathbf{x}$, trained MemDLM p_θ, trained solubility classifier v_ϕ, pre-trained encoder Encoder_ϕ, number of optimization steps N

Produce sequence embeddings $h = \text{Encoder}_\phi(\mathbf{x})$
Compute saliency map $\mathbf{s}$ using gradients $\nabla_h v_\phi(h)$
Normalize saliency map $\hat{s}^i \leftarrow s_i$
Determine editable positions $\mathcal{E}$ based on soluble residues and saliency scores
for each $i \in \mathcal{E}$ **do**
    Define neighborhood $\mathcal{N}(i)$
    Compute $\tilde{s}^i = \hat{s}^i + \gamma \sum_{j \in \mathcal{N}(i)} \text{Norm}(A_{ij}) \cdot \hat{s}^j$
    Construct prior distribution $\pi(x^i)$
    Compute guidance distribution: $\log P(x^i) = (1 - \sigma(\alpha \tilde{s}^i)) \cdot \log p_\theta(x^i) + \sigma(\alpha \tilde{s}^i) \cdot \pi(x^i)$
    Sample token $\hat{x}^i \sim \text{CAT}(\log P(x^i))$
    Update $\mathbf{x}[i] \leftarrow \hat{x}^i$
end for
return Optimized sequence $\hat{\mathbf{x}}$



Path-Planning Sampling Algorithm

Algorithm 2 MemDLM Sampling with P2 Self-Planning and Optional Sequence Refinement

Require: Fully masked sequence $\mathbf{x}_T = \{[\text{MASK}]\}_{i=1}^{L}$, trained MemDLM p_θ, number of denoising steps T

for $t \in \{T, T-1, \ldots, 0\}$ **do**
- Compute logits: $\mathbf{z}_{t-1} = p_\theta(\mathbf{x}_t)$
- Sample candidate tokens: $x_{t-1}^i = \arg\max_v \left(\frac{z_{t-1}^{i,v}}{\tau} + g^{i,v} \right), \quad g^{i,v} \sim \text{Gumbel}(0,1)$
- Compute per-token log-probabilities: $s_t^i = \log p_\theta(x_t^i)$
- Identify unmasked positions: $\mathcal{R}_t = \{i \mid x_{t-1} \neq [\text{MASK}]\}$
- Compute $K = \lfloor (1 - \kappa_t) \cdot |\mathcal{R}_t| \rfloor$
- Select top-K lowest scoring tokens from $\mathcal{R}_t$ and remask them: $x_t^i = [\text{MASK}]$ for $i \in$ top-$K(s_t^i)$
- Copy high-confidence predictions: $x_{t-1}^i \leftarrow x_t^i$ for positions previously masked but not in top-K

end for

if PET Optimization **then**
- Perform Algorithm 3

end if

return Final decoded sequence $\mathbf{x}_0$