Token-Level Guided Discrete Diffusion for Membrane Protein Design

Shrey Goel,[1,2] **Peregrine Michael Schray,**[3] **Yinuo Zhang,**[45] **Sophia Vincoff,**[5]
Huong T. Kratochvil,[3] **Pranam Chatterjee**[2,5,†]

[1]Department of Computer Science, Duke University
[2]Department of Computer and Information Science, University of Pennsylvania
[3]Department of Chemistry, University of North Carolina at Chapel Hill
[4]Duke-NUS Medical School, Singapore
[5]Department of Bioengineering, University of Pennsylvania

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

Abstract

Reparameterized diffusion models (RDMs) have recently matched autoregressive methods in protein generation, motivating their use for challenging design tasks such as membrane proteins, which possess interleaved soluble and transmembrane (TM) regions. We introduce the ***Mem**brane **D**iffusion **L**anguage **M**odel* (**MemDLM**), a fine-tuned RDM-based protein language model that enables controllable membrane protein sequence design. MemDLM-generated sequences recapitulate the TM residue density and structural features of natural proteins, achieving comparable biological plausibility and outperforming state-of-the-art diffusion baselines in motif scaffolding tasks by producing lower perplexity, higher BLOSUM-62 scores, and improved pLDDT confidence. To enhance controllability, we develop **PE**r-**T**oken Guidance (PET), a novel classifier-guided sampling strategy that selectively solubilizes residues while preserving conserved TM domains, yielding sequences with reduced TM density but intact functional cores. Importantly, MemDLM designs validated in TOXCAT β-lactamase growth assays demonstrate successful TM insertion, distinguishing high-quality generated sequences from poor ones. Together, our framework establishes the first experimentally validated diffusion-based model for rational membrane protein generation, integrating *de novo* design, motif scaffolding, and targeted property optimization.

1 Introduction

Membrane proteins play a crucial role in biological systems, regulating molecular transport, signal transduction, and cellular communication [Jelokhani-Niaraki, 2022]. Their capacity to bind specific ligands or undergo conformational changes renders them essential targets for drug development and therapeutics for various diseases [Sanganna Gari et al., 2021]. Even more interestingly, *de novo* design and engineering of membrane proteins offers a powerful therapeutic modality by enabling the creation of highly-specific and stable proteins that can precisely modulate cell signaling pathways, transport processes, and immune responses, making them ideal for targeting diseases such as cancer and neurological disorders [Jelokhani-Niaraki, 2022]. Current methods for designing new protein sequences or scaffolds rely on pre-trained structure prediction networks [Wang et al., 2022, Yin et al., 2007, Elazar et al., 2022], which remains a particularly challenging prerequisite for membrane protein targets. The scarcity of high-resolution structures hinders the training of high-fidelity deep learning

Accepted at the NeurIPS 2025 Workshop on AI for Science (Oral).

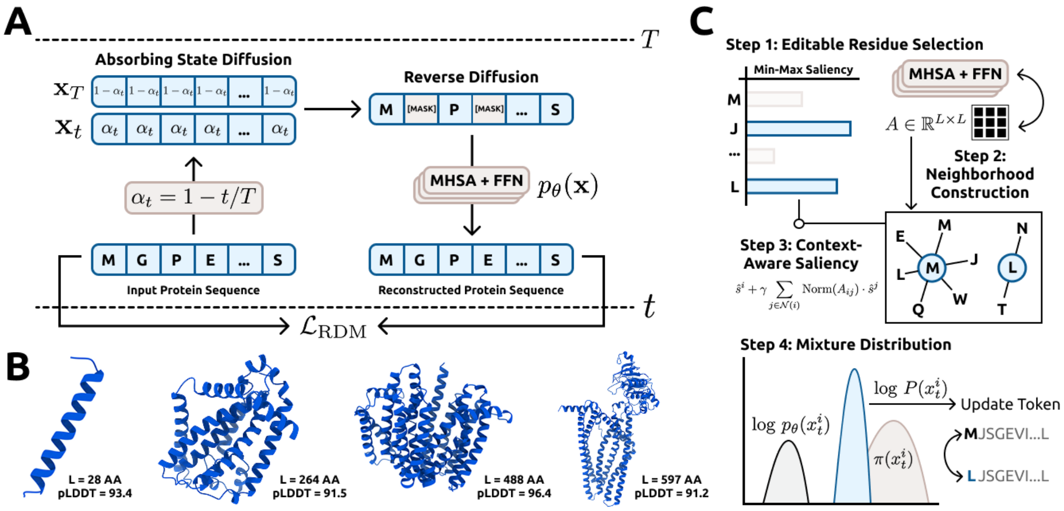


Figure 1: **MemDLM Schematic**. **A)** RDM-based model training diagram. **B)** AlphaFold3 visualizations of unconditional samples. **C**) Token-level classifier guided diffusion sampling with PET algorithm.

structure prediction models for membrane proteins: only $\sim$1% of the current PDB structures are annotated as membrane proteins. Further, energy functions underlying physics-based computational models are suboptimal because they often require iterative optimizations to design analogs of membrane proteins [Vorobieva et al., 2021]. As a result, current methods in *de novo* membrane protein design are limited to simple helical barrel or beta-barrel folds with low sequence complexity.

While deep learning-based topology predictors (e.g., DeepLoc, AllesTM) aid in identifying helix regions and subcellular localization, they primarily analyze existing sequences and do not support *de novo* generation for function-specific design [Thumuluri et al., 2022] [Hönigschmid et al., 2020]. Prior computational design efforts have achieved impressive results by designing zinc-transporting helices, yet they are often limited to fixed scaffolds, small proteins, or require extensive intervention [Joh et al., 2014]. What remains missing is a generative modeling framework that can autonomously produce membrane protein sequences with controllable structural features, including TM helices, soluble domains, and higher-order topologies, without relying on predetermined scaffolds or manual adjustments [Goverde et al., 2024].

In this work, we introduce **MemDLM**, a discrete diffusion protein language model for rational membrane protein design (Figure 1). At the core of our approach is ***PE****r-****T****oken Guidance* (PET), a novel classifier-guided sampling algorithm that combines attention scores and classifier rewards to optimize specific sequence tokens during inference. Unlike traditional classifier-guidance methods ([Gruver et al., 2024], [Li et al., 2024], [Vignac et al., 2022], [Dhariwal and Nichol, 2021], [Tang et al., 2025], [Chen et al., 2025]), PET ensures the retention of targeted tokens, an essential requirement in membrane protein design, where highly conserved transmembrane (TM) domains are critical to maintaining structural topology. We demonstrate that MemDLM generates biologically relevant proteins with structual features resembling membrane proteins (e.g. α-helices) and show that PET solubilizes natural membrane proteins while retaining key functional TM domains. Overall, our integrated pipeline serves as a versatile, end-to-end platform for designing and optimizing membrane protein sequences, with potential applications spanning therapeutics, drug delivery, and synthetic biology.

Our key contributions are as follows:

- We introduce **MemDLM**, a discrete diffusion protein language model specifically fine-tuned for *de novo* generation of membrane protein sequences with controllable structural features.

- We develop PET, a novel classifier-guided sampling algorithm to optimize specific sequence tokens during inference, ensuring the retention of targeted amino acid tokens like conserved TM domains.

- We demonstrate that MemDLM enables controllable sequence generation through token-level editing. In practice, we show MemDLM effectively solubilizes existing natural membrane protein sequences while preserving crucial functional TM regions.
- We motivate MemDLM's utility in real-world therapeutic design by showing it (i) outperforms existing state-of-the-art models by achieving improved sequence-specific computational benchmarks in *de novo* generation and sequence scaffolding tasks, and (ii) produces experimentally validated membrane protein designs that exhibit favorable growth curves under antibiotic selection.

2 Methods

Language Modeling Preliminaries Let $\mathbf{x} = (x^1, x^2, \dots, x^L) \in \{0,1\}^{L \times |\mathcal{V}|}$ denote a discrete sequence of length L, where each token is represented as a one-hot vector over the vocabulary $\mathcal{V} = \{0, 1, \dots, 32\}$. The vocabulary includes 25 canonical and non-canonical amino acids, along with several special tokens [Lin et al., 2023]. *Language modeling* aims to estimate the underlying data distribution $\mathbf{x} \sim q(\mathbf{x})$ using a parameterized probabilistic model $p_\theta(\mathbf{x})$. Since the true distribution $q(\mathbf{x})$ is typically intractable, we approximate it using a neural network with parameters θ. In Supplementary A.1, we lay out the foundation for RDM-based protein language models by considering related modeling paradigms.

2.1 MemDLM

Modeling MemDLM is built on the Reparameterized Diffusion Model (RDM) framework [Zheng et al., 2023]. We define $\textsc{Cat}(x; \mathbf{p})$ as the categorcial distribution on the discrete sequence $\mathbf{x}$ governed by the vector $\mathbf{p} \in \Delta^{|\mathcal{V}|-1}$, where $\Delta^{|\mathcal{V}|-1}$ denotes the $(|\mathcal{V}| - 1)$-dimensional probability simplex. Given a stationary noise distribution $\mathbf{q}_{\text{noise}}$, we define the unconditional prior as $q(\mathbf{x}_t) = \prod_{i=1}^{L} \textsc{Cat}(x_t^i; \mathbf{q}_{\text{noise}})$. We can then write the*forward* diffusion process as a transition kernel defined in closed-form as a convex combination of clean data and noise:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \alpha_t \mathbf{x}_0 + (1 - \alpha_t) q_{noise} \tag{1}$$

where $\alpha_t = \prod_{i=1}^{t} \beta_i = 1 - t/T$ is a linear noise schedule. This transition distribution in Eq. 1 shows that the forward process is ultimately a convex combination of α_t, the probability of clean data $\mathbf{x}_0$ remaining unchanged, and $1 - \alpha_t$, the probability of $\mathbf{x}_0$ transitioning to the $[\text{MASK}]$ token. By sampling $t \sim \mathcal{U}(0, T = 500)$, we can determine the identity of a token at the given timestep of the forward process:

$$x_t^i = \begin{cases} [\text{MASK}] & \text{if } u_i < \frac{t}{T}, \quad u_i \sim \text{Uniform}(0,1) \\ x_0^i & \text{otherwise} \end{cases} \tag{2}$$

Importantly, the forward noising process is characterized by an *absorbing state*: $\lim\limits_{t \to T} \alpha_t = \lim\limits_{t \to T} (1 - t/T) = 0$, indicating all tokens are guaranteed to be replaced by noise. During inference, MemDLM$_\theta$ must *denoise* a fully masked sequence $\mathbf{x}_T = \{[\text{MASK}]\}_{i=1}^{L}$, rendering the absorbing state a necessary ingredient of the forward noising process. In Section 2.2, we formally outline a generalized denoising framework from [Peng et al., 2025] to obtain samples from masked diffusion models (e.g., RDMs).

$$q(\mathbf{x}_t) = \prod_{i=1}^{L} \textsc{Cat}(x_t^i;\, \mathbf{q}_{\text{noise}}) \tag{3}$$

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \alpha_t \, \mathbf{x}_0 + (1 - \alpha_t) \, \mathbf{q}_{\text{noise}}, \qquad \alpha_t = \prod_{i=1}^{t} \beta_i = 1 - \frac{t}{T} \tag{4}$$

$$x_t^i = \begin{cases} [\text{MASK}], & \text{if } u_i < \frac{t}{T}, \quad u_i \sim \text{Uniform}(0,1) \\ x_0^i, & \text{otherwise} \end{cases} \tag{5}$$

$$\lim_{t \to T} \alpha_t = \lim_{t \to T} \left(1 - \frac{t}{T} \right) = 0 \qquad \Rightarrow \qquad \mathbf{x}_T = \{[\text{MASK}]\}_{i=1}^{L} \tag{6}$$

Loss Function Following the proof in [Wang et al., 2024] (Appendix A), the RDM framework simplifies the ELBO (Eq. 17) by breaking down the KL-divergence term to yield a simplified training objective:

$$\begin{aligned} \mathcal{L}_{\text{RDM}} &= -\mathbb{E}_{q(\mathbf{x}_0)} \, \text{KL} \left[q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \right] \\ &= \mathbb{E}_{q(\mathbf{x}_0)} \left[\lambda_t \sum_{i=1}^{L} b^i(t) \cdot \log p_\theta(x_0^i \mid \mathbf{x}_t) \right] \end{aligned} \tag{7}$$

$$\mathcal{L}_{\text{RDM}} = \mathbb{E}_{q(\mathbf{x}_0)} \left[\lambda_t \sum_{i=1}^{L} \mathbf{1}_{x_t^i \neq x_0^i} \cdot \log p_\theta(x_0^i \mid \mathbf{x}_t) \right] \tag{8}$$

where $\lambda_t := T - (t-1)$ represents a linear, time-dependent coefficient and $b^i(t) = \mathbf{1}_{x_t^i \neq x_0^i}$. In practice, $\mathcal{L}_{\text{RDM}}$ can easily be computed using the cross-entropy loss between logits and sequence labels. In Supplementary B.2, we detail the specific architectural and training schemes used to construct MemDLM.

2.2 Path-Planning Sampling

To generate realistic membrane-like protein sequences from a trained MemDLM, we adopt the Path-Planning (P2) paradigm introduced by [Peng et al., 2025], a novel sampling framework for masked discrete diffusion language models. Notably, P2 breaks the assumption of uniform unmasking probabilities and enhances generative quality compared to stochastic sampling from a Gumbel-Softmax distribution or greedy decoding of softmax logits. We follow the *self-planner* variant of P2, where the denoiser itself provides a planning signal used to identify and resample low-value tokens. Here and in Algorithm 2, we outline the key steps of self-planning in P2 but direct the reader to [Peng et al., 2025] for a complete background.

Initial Token Sampling Beginning with a fully masked sequence $\mathbf{x}_t = \{[\text{MASK}]\}_{i=0}^{L}$, MemDLM predicts denoised logits $\mathbf{z}_{t-1} \in \mathbb{R}^{L \times |\mathcal{V}|}$ via $\mathbf{z}_{t-1} = p_\theta(\mathbf{x}_t)$ at each timestep. Candidate tokens are sampled from the logits using Gumbel-softmax decoding with temperature parameter τ:

$$x_{t-1}^i = \arg\max_v \left(\log \text{softmax} \left(\frac{z_{t-1}^{i,v}}{\tau} + g^{i,v} \right) \right), \quad \mathbf{g}_i \sim \text{Gumbel(0, 1)} \tag{9}$$

Self-Planning An important requirement of self-planning is resampling low-value tokens using the predictions of the denoising model. Accordingly, we use MemDLM's log probabilities to compute $s_t^i = \log p_\theta(x_t^i)$, a per-position score, and $\mathcal{R}_t = \mathbf{x}_{t-1}^{\setminus\mathcal{M}}$, the set of unmasked positions $\setminus\mathcal{M}$ eligible for remasking. We select the top-K tokens from $\mathcal{R}_t$ with the lowest log-probability scores s_t^i and remask them. Specifically, we dynamically compute $K = \lfloor (1 - \kappa_t) \cdot |\mathcal{R}_t| \rfloor$ as a fixed proportion of unmasked positions controlled by the monotonic scheduling function $\kappa_t = \kappa\,(i/N)$, where $i \in \{1, 2, \ldots, N\}$ and $\kappa : [0,1] \to [0,1]$. This update forces the token predictions MemDLM was not confident about (low s_t^i) to be remasked.

Token Resampling We sample new tokens at the remasked positions by copying the most recent denoised tokens from the previous timestep $\mathbf{x}_{t-1}$ into the current sequence $\mathbf{x}_t$ at positions that were masked but are no longer among the K lowest-scoring tokens. This step progressively commits high-confidence tokens while leaving low-confidence regions available for further refinement in future steps, a key advantage over ancestral and greedy sampling schemes. By following the self-planning scheme of P2, no additional model training or overhead is required, providing a lightweight inference mechanism for membrane protein design tasks.

2.3 Per-Token Classifier Guided Sampling

While generating arbitrary membrane proteins is valuable, it is insufficient for downstream applications, as unconditional samples are unlikely to exhibit the functional properties required for their use as therapeutic modalities [Jelokhani-Niaraki, 2022]. *Classifier-guided sampling* has recently introduced controllability to deep generative models by following a gradient signal from a pre-trained classifier model [Gruver et al., 2024], [Li et al., 2024], [Vignac et al., 2022], [Dhariwal and Nichol, 2021], [Tang et al., 2025], [Chen et al., 2025]. Although these methods bias the model's sampling trajectory towards the desired class label, there is no guarantee that specific sequence tokens are preserved during inference.

To this end, we introduce ***Per-Token Guidance*** (PET), a novel classifier-guided sampling algorithm that selects and replaces specific sequence tokens with optimized analogues, moving the overall sequence towards the desired property (Figure 1C). In the case of membrane protein design, PET can readily be used to replace noncritical TM residues with soluble analogues to guarantee overall sequence solubility while maintaining biologically conserved TM domains. Solubilizing membrane proteins without disrupting these critical TM residues is essential for ensuring functional foldability and membrane localization, as TM residues often mediate key structural and biophysical interactions. Below, we carefully outline our PET algorithm and refer the reader to Supplementary A.2 for a background on discrete classifier guidance.

Setup Given a sequence consisting of only amino acid tokens, $\mathbf{x} = \{x_i \in \text{Canonical}\}_{i=1}^{L}$, PET first identifies a dynamic subset of editable positions $\mathcal{E} \subseteq \{1, \dots, L\}$ using existing residue annotations or a trained per-token solubility classifier $v_\phi : \mathbb{R}^{B\times L\times D} \to \mathbb{R}^{B\times L}$. This classifier operates over the hidden states h derived from the ESM-2-650M protein language model [Lin et al., 2023] and is trained on fully unmasked sequences. See Section B.3 for full training details regarding v_ϕ.

Determining Editable Residues PET first constructs a set of conserved, *non-editable* token indices $\mathcal{C}$ based on solubility annotations or predictions:

1. If soluble residue annotations $\mathcal{S} \subseteq \{1, 2, \dots, L\}$ are provided (e.g. experimentally-derived labels for known membrane protein sequences), initialize $\mathcal{C} = \mathcal{S}$.
2. If no annotations are provided, initialize $\mathcal{C} = \{i \in \{1, \dots, L\} \mid v_\phi(h_t)_i \geq 0.5\}$. Inherently, it is assumed that some $v_\phi(h_t)_i < 0.5$.

Next, we consider low-value tokens, *i.e.*, insoluble amino acids with TM-like character. It is critical to maintain the most conserved TM regions during optimization to maintain the biological plausibility of the membrane protein. Thus, we guide the selection of *unimportant* TM residues under LaMBO-2's (Supplementary A.2) definition of a token's *saliency* $s^i(h)$, a score that quantifies a token's importance relative to the classifier v_ϕ [Gruver et al., 2024]. Given a sequence's latent representation, we construct a *saliency map* $\mathbf{s} = (s^1, s^2, \dots, s^L) \in \mathbb{R}^L$:

$$\mathbf{s}(h) := \max\Bigg\{ \Bigg(\sum_{d=1}^{D} \big|\nabla_h v_\phi(h)_d\big| \Bigg)^{1/\tau}, \quad \epsilon \Bigg\}, \quad \hat{s}^i := \frac{s^i - \min \mathbf{s}}{\max \mathbf{s} - \min \mathbf{s} + \delta} \tag{10}$$

using temperature $\tau = 2.0$ and a ceiling $\epsilon = e^{-4}$ to stabilize gradient noise. Although LaMBO-2 normalizes the saliency map to the probability distribution $P_{\text{edit}}(\mathbf{x}_t) = \mathbf{s}/\sum_i s_i$ ([Gruver et al., 2024], Eq. 5), PET opts for min-max scaling (Eq. 10) to prevent vanishing probabilities for large L. If v_θ is well-trained, high values of $\mathbf{s}$ should correlate with low-value (TM-like) residues. To finalize $\mathcal{C}$, PET selects the top-K most salient tokens:

$$\mathcal{C} = \mathcal{C} \cup \text{top-}K\big(\hat{\mathbf{s}}, K = \max\big\{1, \tfrac{1}{10} \cdot (L - |\mathcal{C}|)\big\}\big), \quad \mathcal{E} = \{1, \dots, L\} \setminus \mathcal{C} \tag{11}$$

Together, these token selection strategies define $\mathcal{E}$, the set of editable token indices. This set excludes soluble and highly salient residues to preserve membrane protein character (TM-like residues) while optimizing for sequence solubility.

Neighborhood Construction. For each editable token $i \in \mathcal{E}$, PET constructs a context-aware *neighborhood* $\mathcal{N}(i)$ based on attention scores. Let $A \in \mathbb{R}^{L \times L}$ be the final-layer attention matrix extracted from p_θ. The neighborhood $\mathcal{N}(i)$ is formed using top-p nucleus sampling over the normalized attention weights $\text{Norm}(A_{i,:}/\tau)$, excluding special tokens and the self-position i; we set $\tau = 1/\log L$ to ensure neighborhood selection is neither overly diffuse in long sequences nor overly narrow in short sequences. Thus, the final neighborhood contains all tokens j such that the cumulative attention probability $\sum_{j' \in \mathcal{N}(i)} A_{ij'}$ exceeds the threshold $p = 0.9$. The construction of an attention-informed neighborhood is necessary to propagate long-range residue information to avoid blindly modifying individual tokens.

Context-Aware Saliency PET then refines a token's raw saliency score s_i with contributions from the token's attention-weighted neighborhood $\mathcal{N}(i)$. The *context-aware saliency* score $\tilde{s}^i$ is defined as:

$$\tilde{s}^i := \hat{s}^i + \gamma \sum_{j \in \mathcal{N}(i)} \frac{A_{ij}}{\sum_{j' \in \mathcal{N}(i)} A_{ij'}} \cdot \hat{s}^j \tag{12}$$

where $\gamma = 0.5$ controls the influence of the neighborhood saliency. Overall, the context-aware saliency blends both the intrinsic importance of the token x^i with the contributions of tokens it attends to most strongly, creating a holistic representation of an individual residue's contribution to sequence-level solubility.

Mixture Distribution Let $\log p_\theta(x_t^i)$ be the log-probability distribution across the vocabulary for a singular token by the language model at timestep t, and let $\pi(x_t^i)$ be a prior token distribution in log-space. To update a token, PET defines a *mixture distribution* $\log P(x_t^i)$ for each editable position $i \in \mathcal{E}$:

$$\log P(x_t^i) = (1 - w^i) \cdot \log p_\theta(x_t^i) + w^i \cdot \pi(x_t^i) \tag{13}$$

By construction, $P(x_t^i)$ remains a valid probability distribution, as it is a convex combination of two normalized distributions. The mixture weight w^i can be computed as:

$$w_i = \sigma(\alpha \cdot \tilde{s}^i) \tag{14}$$

with $\sigma(\cdot)$ denoting the sigmoid function and $\alpha = 5.0$ controlling the sharpness of the transition. Eq. 13 ensures that an updated token's distribution is biased towards the prior when $\tilde{s}^i$ is large since $s_i \to 1$ when $v_\theta(h_t^i) \to 0$. Biologically, this corresponds to a residue with high TM-like character that is thus conserved and should remain fixed. Conversely, when $\tilde{s}^i$ is small, PET favors the model's default prediction, allowing more flexibility in low-saliency (non-critical) positions.

Prior Distribution In order to consutrct the mixture distribution, we define a *temporal prior* $\pi(x_t^i) := \log p_\theta(x_{t-1}^i)$ in PET sampling that leverages the denoising model's log probabilities from a previous diffusion timestep. This formulation maintains the likelihood of the original sequence while encouraging updates from the mixture weighting in Eq. 13.

Token Sampling and Preservation. A new token $\hat{x}^i$ is sampled from $P(x^i)$ for each position $i \in \mathcal{E}$. By design, PET will not update positions $j \notin \mathcal{E}$, resulting in an optimized sequence that preserves soluble and conserved TM regions while refining low-saliency, TM positions. To produce optimized amino acid tokens, we sample from a categorical distribution parameterized by the updated token probabilities at each position, $\hat{x}^i \sim \textsc{Cat}(\log P(x^i))$.

2.4 TOXCAT-β-Lactamase Growth Assay

The TOXCAT-β-lactamase assay was used to evaluate membrane insertion and TM association of MemDLM-generated sequences [Russ and Engelman, 1999, Lis and Blumenthal, 2006]. Candidate designs were cloned between an N-terminal ToxR transcriptional activator and a C-terminal periplasmic β-lactamase in the pMAL_dstβL vector, and transformed into *E. coli* Cloni cells. Single colonies were used to inoculate LB cultures with spectinomycin, diluted to OD_{600} = 0.05, and normalized to 1.95×10^5 cells per well in 96-well plates. Cultures were grown in LB supplemented with spectinomycin (50 μg/mL) and subjected to different selective pressures: carbenicillin (300 μg/mL) to report on membrane insertion, or combined carbenicillin (100 μg/mL) and chloramphenicol (100–120 μg/mL) to report on TM-mediated oligomerization. Plates were incubated at 37°C with continuous shaking in a BioTek Synergy H1 plate reader, and growth was monitored by OD_{600} every 10 minutes for 24 hours. Successful insertion positions β-lactamase in the periplasm to hydrolyze carbenicillin, while oligomerization activates the *ctx* promoter via ToxR dimerization, conferring chloramphenicol resistance.

3 Results

3.1 *De Novo* Generation

Given the limited availability of experimentally verified membrane structures, we focused on sequence-based metrics (Supplementary B.4). Notably, we computed the TM Residue Density of the generated sequences by predicting TM and soluble residue regions with DeepTMHMM [Hallgren et al., 2022]. To realize this comparison, we utilized all 1,098 sequences from the MemDLM model test set as the basis for our experiments, yielding a realistic evaluation of sequence plausability and membrane character.

	pLDDT (↑)	TM Residue Density	PPL (↓)	Entropy (↑)
Test Set	76.637	0.294	5.707	3.918
MemDLM	67.410	0.311	6.344	3.743

Table 1: Computational validation of generated and experimentally validated membrane proteins

Table 1 compares various metrics of experimentally annotated membrane proteins with *de novo*-generated sequences. The results show that MemDLM generates sequences with a soluble residue density closely matching that of experimentally verified membrane proteins, indicating that MemDLM has successfully learned their underlying distribution (Supplementary A1).

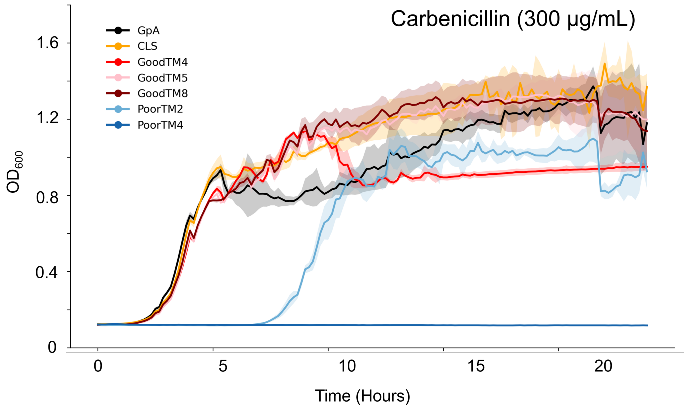


Figure 2: Growth curves of MemDLM-generated TM sequences under carbenicillin (300 μg/mL).

To fully validate MemDLM's *de novo* generative capabilities, we selected three generated sequences considered to be *single-pass* membrane proteins ("GoodTM") from the top-100 and two from the bottom-22 ("PoorTM") set of MemDLM-generated sequences for experimental validation in *Escherichia coli* (*E. coli*) using TOXCAT-β-lactamase bacterial growth assays [Lis and Blumenthal, 2006], which employ a dual-reporter system for evaluating membrane insertion and oligomerization of single-pass peptides and proteins [Russ and Engelman, 1999, Lis and Blumenthal, 2006, Ottemann and Mekalanos, 1995, Armstrong and Senes, 2016, Elazar et al., 2016] (Supplementary B.5, C.3). In these constructs, the design of interest is inserted between an N-terminal ToxR cytoplasmic domain and a C-terminal periplasmic β-lactamase. *E. coli* survival under different antibiotic selection pressures then provides a direct functional readout: survival in carbenicillin indicates successful membrane insertion, which positions the β-lactamase in the periplasm to degrade the antibiotic, while growth in carbenicillin and chloramphenicol demonstrates TM-mediated oligomerization, where multimerization of the ToxR transcription factors activates the downstream ctx promoter that confers resistance to chloramphenicol.

Figure 2 shows the TOXCAT growth curves for poor and high-quality MemDLM sequences alongside the positive insertion controls GpA and CLS (Supplementary A5). Under carbenicillin selection (300 μg/mL), GpA, CLS, GoodTM4, GoodTM5, and GoodTM8 all achieved similar growth kinetics and reached the midpoint of log-phase growth at $\sim$4 hours, demonstrating similar membrane insertion efficiencies. PoorTM4 showed no growth in carbenicillin, much like our negative controls (Supplementary A6), indicating that the sequence is not membrane-inserting. However, PoorTM2, which contains six charged residues within the predicted TM span, also grew in carbenicillin but with a noticeable delay, suggesting weaker membrane insertion propensity. The survival of GoodTM designs under carbenicillin selection demonstrates that MemDLM can generate *de novo* TM-inserting sequences and that filtering generated sequences with computational metrics effectively ranks TM-like sequences. The poor survivability of PoorTM2 and PoorTM4, both ranked among the bottom 22 sequences by MemDLM, compared to the GoodTM designs further supports MemDLM's ability to distinguish TM-like sequences.

3.2 Motif Scaffolding

As a natural extension of *de novo* design, we scaffolded around TM and soluble motifs of experimentally annotated membrane proteins. We take the entire test set, comprising 1,098 experimentally verified membrane protein sequences with annotated TM and soluble motifs, and mask out all residues except those in the TM or soluble motif(s). We use these partially masked sequences as input to the models to assay their capability to generate scaffolds conditioned on known TM or soluble motifs. We focused on these domains due to their distinct hydrophilic and hydrophobic regions that govern the folding and thus function of the overall protein.

	pLDDT (↑)		PPL (↓)		BLOSUM-62 (↑)		ENTROPY (↑)	
	INSOL	SOL	INSOL	SOL	INSOL	SOL	INSOL	SOL
Test Set	76.637	76.637	5.707	5.707	–	–	3.918	3.918
EvoDiff	**64.058**	64.036	9.841	4.632	2.176	-0.188	3.841	**3.841**
MemDLM	62.762	**70.112**	**8.748**	**3.242**	**2.964**	**0.512**	**3.876**	3.803

Table 2: Reconstruction quality comparison of models scaffolding around TM and soluble motifs of 1,098 experimental membrane protein sequences that represent the MemDLM model test set.

Our results (Table 2, Supplementary A2, A3) show that MemDLM-inpainted sequences achieve lower average pseudo-perplexities and higher pLDDT and BLOSUM-62 scores relative to EvoDiff-based ([Alamdari et al., 2023]) scaffolds. These results suggest that MemDLM scaffolds functional motifs with greater confidence while preserving biological relevance compared to SOTA diffusion models.

3.3 Solubilizing Targeted Residues

Finally, we apply PET to optimize specific residues spanning the insoluble regions of the test set proteins, observing a decrease in TM Residue Density while still preserving critical TM domains (Table 3, Supplementary A4).

	pLDDT (↑)	TM RESIDUE DENSITY (↓)	PPL (↓)	BLOSUM-62 (↑)	ENTROPY (↑)
Test Set	76.637	0.294	5.707	–	3.918
MemDLM	62.979	0.181	8.472	0.495	3.870

Table 3: Computational validation of membrane proteins solubilized under the PET sampling strategy.

As a final validation, we visualize MemDLM-generated sequences with AlphaFold3 (Supplementary D) and confirm the presence of hallmark membrane protein structures, including α-helical bundles and distinct TM and soluble regions [Zhang et al., 2015].

4 Discussion

In this work, we introduce MemDLM, the first classifier-guided masked diffusion language model designed specifically for *de novo* membrane protein generation. By leveraging the strengths of masked diffusion over traditional structure-based models, MemDLM effectively captures long-range dependencies critical to the structural and functional integrity of membrane proteins – an area where structure-based models often fall short due to their reliance on pre-defined structural templates and limited generation across diverse topologies. Furthermore, our integration of Per-Token Guidance (PET) for classifier-guided sampling further enables property-guided optimization, enabling us to generate soluble residues over existing TM domains while retaining an initial sequence scaffold. MemDLM also outperforms existing models at demonstrating a robust capability in scaffolding functional motifs, maintaining biological relevance, and achieving high similarity to natural proteins. Moving forward, we aim to generate diverse membrane topologies, including β-barrel and higher-order states and continue to experimentally characterize MemDLM-generated membrane proteins. By evaluating the structural and functional properties of scaffolded TM domains and testing the solubility and stability of membrane proteins generated through classifier-guided optimization, we will validate MemDLM's potential for advancing rational membrane protein design and expanding its applications in therapeutic development.

References

Masoud Jelokhani-Niaraki. Membrane proteins: structure, function and motion. *International journal of molecular sciences*, 24(1):468, 2022.

Raghavendar Reddy Sanganna Gari, Joel José Montalvo-Acosta, George R Heath, Yining Jiang, Xiaolong Gao, Crina M Nimigean, Christophe Chipot, and Simon Scheuring. Correlation of membrane protein conformational and functional dynamics. *Nature Communications*, 12(1):4363, 2021.

Jue Wang, Sidney Lisanza, David Juergens, Doug Tischer, Joseph L Watson, Karla M Castro, Robert Ragotte, Amijai Saragovi, Lukas F Milles, Minkyung Baek, et al. Scaffolding protein functional sites using deep learning. *Science*, 377(6604):387–394, 2022.

Hang Yin, Joanna S. Slusky, Bryan W. Berger, Robin S. Walters, Gaston Vilaire, Rustem I. Litvinov, James D. Lear, Gregory A. Caputo, Joel S. Bennett, and William F. DeGrado. Computational design of peptides that target transmembrane helices. *Science*, 315(5820):1817–1822, March 2007. ISSN 1095-9203. doi: 10.1126/science.1136782. URL `http://dx.doi.org/10.1126/science.1136782`.

Assaf Elazar, Nicholas J Chandler, Ashleigh S Davey, Jonathan Y Weinstein, Julie V Nguyen, Raphael Trenker, Ryan S Cross, Misty R Jenkins, Melissa J Call, Matthew E Call, and Sarel J Fleishman. De novo-designed transmembrane domains tune engineered receptor functions. *eLife*, 11, May 2022. ISSN 2050-084X. doi: 10.7554/elife.75660. URL `http://dx.doi.org/10.7554/eLife.75660`.

Anastassia A Vorobieva, Paul White, Binyong Liang, Jim E Horne, Asim K Bera, Cameron M Chow, Stacey Gerben, Sinduja Marx, Alex Kang, Alyssa Q Stiving, et al. De novo design of transmembrane β barrels. *Science*, 371(6531):eabc8182, 2021.

Vineet Thumuluri, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Henrik Nielsen, and Ole Winther. Deeploc 2.0: multi-label subcellular localization prediction using protein language models. *Nucleic acids research*, 50(W1):W228–W234, 2022.

Peter Hönigschmid, Stephan Breimann, Martina Weigl, and Dmitrij Frishman. Allestm: predicting multiple structural features of transmembrane proteins. *BMC bioinformatics*, 21(1):242, 2020.

Nathan H Joh, Tuo Wang, Manasi P Bhate, Rudresh Acharya, Yibing Wu, Michael Grabe, Mei Hong, Gevorg Grigoryan, and William F DeGrado. De novo design of a transmembrane zn2+-transporting four-helix bundle. *Science*, 346(6216):1520–1524, 2014.

Casper A Goverde, Martin Pacesa, Nicolas Goldbach, Lars J Dornfeld, Petra EM Balbi, Sandrine Georgeon, Stéphane Rosset, Srajan Kapoor, Jagrity Choudhury, Justas Dauparas, et al. Computational design of soluble and functional membrane protein analogues. *Nature*, 631(8020):449–458, 2024.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36, 2024.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *ArXiv*, pages arXiv–2412, 2025.

Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow matching for controllable biological sequence design. *arXiv preprint arXiv:2505.07086*, 2025.

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.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024.

William P Russ and Donald M Engelman. Toxcat: a measure of transmembrane helix association in a biological membrane. *Proceedings of the National Academy of Sciences*, 96(3):863–868, 1999.

Maciej Lis and Kenneth Blumenthal. A modified, dual reporter toxcat system for monitoring homodimerization of transmembrane segments of proteins. *Biochemical and biophysical research communications*, 339(1):321–324, 2006.

Jeppe Hallgren, Konstantinos D Tsirigos, Mads Damgaard Pedersen, José Juan Almagro Armenteros, Paolo Marcatili, Henrik Nielsen, Anders Krogh, and Ole Winther. Deeptmhmm predicts alpha and beta transmembrane proteins using deep neural networks. *BioRxiv*, pages 2022–04, 2022.

Karen M Ottemann and John J Mekalanos. Analysis of vibrio cholierae toxr function by construction of novel fusion proteins. *Molecular microbiology*, 15(4):719–731, 1995.

Claire R Armstrong and Alessandro Senes. Screening for transmembrane association in divisome proteins using toxgreen, a high-throughput variant of the toxcat assay. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 1858(11):2573–2583, 2016.

Assaf Elazar, Jonathan Weinstein, Ido Biran, Yearit Fridman, Eitan Bibi, and Sarel Jacob Fleishman. Mutational scanning reveals the determinants of protein insertion and association energetics in the plasma membrane. *Elife*, 5:e12125, 2016.

Sarah Alamdari, Nitya Thakkar, Rianne Van Den Berg, Neil Tenenholtz, Bob Strome, Alan Moses, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *BioRxiv*, pages 2023–09, 2023.

Shao-Qing Zhang, Daniel W Kulp, Chaim A Schramm, Marco Mravic, Ilan Samish, and William F DeGrado. The membrane-and soluble-protein helix-helix interactome: similar geometry via different interactions. *Structure*, 23(3):527–541, 2015.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Sophia Vincoff, Shrey Goel, Kseniia Kholina, Rishab Pulugurta, Pranay Vure, and Pranam Chatterjee. Fuson-plm: a fusion oncoprotein-specific language model via adjusted rate masking. *Nature Communications*, 16(1):1436, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Andrei L. Lomize, Spencer C. Todd, and Irina D. Pogozheva. Spatial arrangement of proteins in planar and curved membranes by <scp>ppm</scp> 3.0. *Protein Science*, 31(1):209–220, November 2021. ISSN 1469-896X. doi: 10.1002/pro.4219. URL `http://dx.doi.org/10.1002/pro.4219`.

Mikhail A Lomize, Andrei L Lomize, Irina D Pogozheva, and Henry I Mosberg. Opm: orientations of proteins in membranes database. *Bioinformatics*, 22(5):623–625, 2006.

Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

Marcin Wolny, Matthew Batchelor, Gail J Bartlett, Emily G Baker, Marta Kurzawa, Peter J Knight, Lorna Dougan, Derek N Woolfson, Emanuele Paci, and Michelle Peckham. Characterization of long and stable de novo single alpha-helix domains provides novel insight into their stability. *Scientific reports*, 7(1):44341, 2017.

A Wettig, T Gao, Z Zhong, and D Chen. Should you mask 15% in masked language modeling? arxiv 2022. *arXiv preprint arXiv:2202.08005*.

Renhao Li, Roman Gorelik, Vikas Nanda, Peter B Law, James D Lear, William F DeGrado, and Joel S Bennett. Dimerization of the transmembrane domain of integrin αiib subunit in cell membranes. *Journal of Biological Chemistry*, 279(25):26666–26673, 2004.

William P Russ and Donald M Engelman. The gxxxg motif: a framework for transmembrane helix-helix association. *Journal of molecular biology*, 296(3):911–919, 2000.

A Extended Background

A.1 Language Modeling

Masked Language Models Masked Language Models (MLMs) employ Transformer-based architectures to learn bi-directional sequence context, distant token relationships, and predict the identity of corrupted (masked) amino acid tokens. The model is trained under a sequence-recovery training objective:

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \mathcal{M}} \log p_\theta(x^i | x^{\backslash \mathcal{M}}) \tag{15}$$

where the set of masked positions $\mathcal{M}$ is a fraction of the sequence tokens. MLMs are strong representation-learners and excel at understanding both protein and natural languages. However, training these models to reconstruct only a minor fraction of tokens (15-40%) across a sequence makes complete *de novo* sequence generation difficult. [Devlin, 2018] [Lin et al., 2023] [Vincoff et al., 2025].

Autoregression AR language models apply the chain rule to obtain a sequential factorization. These models are trained to maximize the log-likelihood of the data:

$$\mathbb{E}_{q(\mathbf{x})} \log p_\theta(\mathbf{x}) = \mathbb{E}_{q(\mathbf{x})} \sum_{i=1}^{L} \log p_\theta(\mathbf{x}^i | \mathbf{x}^{1:L}) \tag{16}$$

New samples can be drawn ancestrally in L steps ($x^1 \sim p_\theta(x^1), \ldots, x^L \sim p_\theta(x^L | x^{1:L-1})$) following a strictly left-to-right unidirectional protocol. These models are a viable choice for natural language modeling schemes where a linear relationship between past and present values is inherently assumed. However, in biological contexts, such as protein sequences, AR models are limited by their inability to capture non-linear and long-range dependencies. For example, multi-pass membrane proteins consist of interleaved TM and soluble regions that are spatially and functionally coupled but may be separated by long sequence distances.

Denoising Diffusion Models Diffusion models are a class of generative models defined by Markov processes [Ho et al., 2020] [Sohl-Dickstein et al., 2015]. The *forward* diffusion steps $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$ progressively corrupt an initial data sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ into a noisy prior $\mathbf{x}_T \sim q_{\text{noise}}$ across T timesteps. The noise distribution q_{noise} typically corresponds to an isotropic Gaussian, $\mathcal{N}(0, I)$, in continuous latent spaces, or a uniform categorical distribution over the vocabulary, Cat($|\mathcal{V}|$), in the discrete case. During inference, the learned *backward* process $p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_t) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ gradually denoises the corrupted data sample to obtain samples from the true data distribution. Diffusion models are trained to maximize the evidence lower bound (ELBO):

$$\begin{aligned}\mathbb{E}_{q(\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0)\right] &\geq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right] \\ &= \mathbb{E}_{q(\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1) + \text{const.} - \sum_{t=2}^{T} \underbrace{\text{KL}\left(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\right)}_{\mathcal{F}_t}\right]\end{aligned} \tag{17}$$

New data samples can be drawn by sampling from $q_{\text{noise}}(\mathbf{x}_T)$ and iteratively applying the learned denoising process $p_\theta(\mathbf{x}_{t-1}) = p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Various authors ([Sahoo et al., 2024], [Zheng et al., 2023]) have made simplifying assumptions about the reverse process to derive a computationally inexpensive loss function that reduces to a weighted negative log-likelihood, akin to a weighted form of Eq. 15.

A.2 Classifier-Guided Sampling

Preliminaries Given a property y, guided diffusion aims to maximize $q(y|\mathbf{x})$ by sampling from the joint distribution $\mathbf{x} \sim q(\mathbf{x}_0, y)$. Therefore, the reverse transition can be conditioned on the property value y and prior sequence samples. Using Bayes theorem, the conditional joint distribution can be decomposed:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \frac{q(y|\mathbf{x}_{t-1}, \mathbf{x}_t)}{q(y|\mathbf{x}_t)} \tag{18}$$

In practice, the true distribution of $q(y|\mathbf{x}_t)$ is unknown and can be learned with a neural network $p_\phi(y|\mathbf{x}_t)$. To yield a tractable marginal reverse transition from Eq. 18, we can substitute the true distribution $q(\cdot)$ with our learned neural networks:

$$p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \frac{p_\theta(y|\mathbf{x}_{t-1}, \mathbf{x}_t)}{p_\phi(y|\mathbf{x}_t)} \tag{19}$$

The normalization term in the denominator $p_\phi(y|\mathbf{x}_t)$ can be safely dropped since the model's parameters learn the normalized distribution. We can update the parameters θ, ϕ at each iteration in the direction given by the gradient

$$\nabla_{\mathbf{x}_{t-1}} \log p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = \nabla_{\mathbf{x}_{t-1}} \log p_\phi(y|\mathbf{x}_{t-1}) + \nabla_{\mathbf{x}_{t-1}} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{20}$$

With this formulation, we can steer the denoising trajectory of the unconditional diffusion model to maximize the target attribute y using gradients from an external classifier [Dhariwal and Nichol, 2021]. Unlike classifier-free guidance, classifier-guidance prevents expensive retraining of existing denoising network on high-quality, task-specific labeled data and opens avenues for flexible, plug-and-play conditioning for various downstream applications.

Discrete Classifier Guidance While classifier guidance is well-formulated for diffusion models that operate over continuous data in Euclidean space [Dhariwal and Nichol, 2021], applying it to discrete spaces requires additional approximation. One common approach treats discrete tokens as continuous relaxations on the probability simplex and uses a first-order Taylor expansion around $\mathbf{x}_t$ to approximate $\log p_\phi(y|\mathbf{x}_{t-1})$ by making $\nabla_{\mathbf{x}_t}(\cdot)$ a valid operator. However, this approximation can be inaccurate when the local linearization poorly captures the classifier's behavior over discrete transitions, especially in regions with sharp decision boundaries. To remedy this, several methods ([Li et al., 2024], [Vignac et al., 2022]) have been proposed to circumvent the lack of continuous representations in discrete gradient guidance; most relevant to our work is LaMBO-2 introduced by [Gruver et al., 2024].

LaMBO-2 To realize classifier-guidance for discrete sequences, LaMBO-2 first conducts sequence optimization using a Langevin process over a property-informed latent space. We begin with the discrete Langevin dynamics used in score-based models:

$$\mathbf{x}'_t = \mathbf{x}_t - \eta \nabla_{\mathbf{x}} \log p_\theta(y \mid \mathbf{x}_t) + \sqrt{2\eta\tau}, \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \tag{21}$$

and generalize this update to the continuous latent space $h'_t \in \mathbb{R}^{1\times D}$ guided by a differentiable surrogate of the discrete generative model. The batch size dimension B is set to 1 for simplicity. The latent update step is defined as:

$$h'_t \leftarrow h'_t - \eta \nabla_{h'_t} [\lambda \mathrm{KL}(p_\theta(\mathbf{x}_t|h'_t) \,||\, p_\theta(\mathbf{x}_t|h_t)) - \sigma(v_\theta(h'_t)_d)] + \sqrt{2\eta\tau}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I) \tag{22}$$

with step size η, temperature τ, and regularization strength λ, where the sigmoid operator $\sigma(\cdot)$ can be applied to produce a sequence-level binary class probability from the classifier's unnormalized logit. The explore-exploit loss $\mathcal{L}_{\text{EE}} := \lambda[\mathrm{KL}(p_\theta(\mathbf{x}_t|h'_t) \,||\, p_\theta(\mathbf{x}_t|h_t)] - \sigma(v_\theta(h'_t)_d)$ guides the latent representation towards high values of the property with the gradient $\nabla_h \sigma(v_\theta(h))$, while the KL term ensures the transition distribution maximizes the original sequence likelihood. Given a discrete sequence $\mathbf{x}_t$ and its corresponding latent representation h_t, one can take N Langevin steps of Eq. 22 to realize optimized sequence latent representations before using the language-modeling head of the denoising network to project continuous embeddings to the discrete logit space ([Gruver et al., 2024], Appendix B.2). However, this construction does not guarantee the retention of specific tokens during inference because even if gradients are suppressed for particular positions, the subsequent projection through the language modeling head back into discrete logits does not ensure that the tokens with minimal gradient updates will be preserved.

B Extended Methods

B.1 Dataset Curation

MemDLM Bioassembly structures from X-ray scattering or electron microscopy with better than 3.5 Å resolution, annotated by PDBTM1, mpstruc2, OPM3, or MemProtMD4, were used to curate membrane protein sequences for fine-tuning. *de novo* designed membrane proteins were added manually to the database. The proteins were culled at 100% sequence identity and 30% sequence identity to result in a non-redundant set and a sequence-diverse set, respectively. Integral membrane residues, defined as residues with at least one atom within the bilayer, were parsed from the resulting bioassembly structures using the membrane boundaries predicted by PPM 3.0 [Lomize et al., 2021]. From the dataset of integral membrane residues, only structures with at least one TM chain spanning the entire membrane bilayer were included in the dataset. Additionally, chains without integral membrane residues were removed from the structure. All peripheral membrane proteins, defined as proteins with no TM chain, were filtered out. The TM protein sequences at the two sequence identity cut-offs and the Python script that parses the sequences from the PPM predictions are included in the SI. After these steps, 9,329 sequences with corresponding per-residue annotations remained. To augment this set of sequences, we obtained 2,579 unique PDB IDs from the Orientations of Proteins in Membranes (OPM) database with the provided "subunits" file [Lomize et al., 2006]. PDB IDs were converted to corresponding protein sequences and per-residue labels (TM or soluble) were assigned using the subunits file. The final set of 11,908 TM sequences were then split using the MMSeqs2 easy clustering module with a minimum sequence identity of 80% and a coverage threshold of 50%. The resulting clusters were split to an 80-10-10 ratio into the training set (9,802 sequences, 82.31%), the validation set (1,008 sequences, 8.47%), and the testing set (1,098 sequences, 9.22%).

PET Sampling Classifier We leveraged the same train/test/val set of 11,908 membrane sequences from the MemDLM dataset to develop a binary classifier that predicts the solubility of each amino acid within a protein sequence. Each sequence was annotated on a per-residue basis, with TM (class 1) and soluble (class 0) labels assigned according to the sequence's uppercase and lowercase residues, respectively.

B.2 Modeling MemDLM

Model Architecture EvoFlow is a protein language model consisting of 33 Transformer-encoder layers and a language modeling head that is capable of *de novo* generating protein sequences. More formally, it can denoise a protein sequence consisting of all [MASK] tokens, making it a natural choice for a discrete diffusion-based protein language model. We use the pre-trained EvoFlow protein language model checkpoint (`https://huggingface.co/fredzzp/EvoFlow-650M-context-3070`) as the basis of our neural network p_θ since EvoFlow was trained under the RDM framework (forward process as defined by Eq. 1 and loss computation defined by Eq. 8). The Diffusion Protein Language Model (DPLM) was also trained under the RDM framework by [Wang et al., 2024] and is thereby an alternative choice for p_θ. However, we opt for EvoFlow over DPLM as the architecture for p_θ as DPLM is restricted by its shorter context length of 1,024 tokens, compared to EvoFlow's extended context length of 3,070 tokens.

Training To achieve membrane protein-specific generation, we fine-tuned EvoFlow by selectively updating a subset of the encoder's attention layers. Specifically, the final $N = 3$ Transformer encoder layers $\{\mathcal{L}_{M-N+1}, \ldots, \mathcal{L}_M\}$ are partially unfrozen, where $M = 33$ is the total number of encoder layers. Within each layer, we enable gradient updates to only the key, query, and value projection matrices (W_K, W_Q, and W_V) of the self-attention mechanism and keep all other weights frozen. With this training recipe, we bias the pre-existing EvoFlow latent space with physicochemical features of membrane proteins without overfitting on the new sequences. MemDLM was trained to minimize the objective in Eq. 8 on a 4xA6000 NVIDIA DGX server with 200 GB of shared VRAM for 3K steps using the AdamW optimizer (betas=($\beta_1 = 0.99, \beta_2 = 0.98$), weight decay $\lambda = 0.01$), a learning rate (LR) of 4×10^{-5} with a cosine schedule (150 warmup steps, LR minimum = 1×10^{-5}).

B.3 Per-Token Solubility Classifier

Let $v_\phi : \mathbb{R}^{B \times L \times D} \to \mathbb{R}^{B \times L}$ be a neural network trained to predict per-token solubility scores from continuous latent representations h_t. The model is trained using clean protein sequences $\mathbf{x}$ with corresponding binary per-residue solubility labels $\mathbf{y} \in \{0, 1\}^L$ (0 = insoluble, 1 = soluble). Each input sequence is first embedded using the pretrained ESM-2-650M protein language model checkpoint (`https://huggingface.co/facebook/esm2_t33_650M_UR50D`) [Lin et al., 2023]. The resulting contextualized token embeddings are passed through a lightweight classifier v_ϕ with the following architecture: (i) trainable 2-layer Transformer encoder Transformer$_\phi$; (ii) LayerNorm and dropout ($p = 0.5$); and (iii) a trainable 2-layer projection head MLP$_\phi$ outputs a scalar logit for each token position. All parameters in ESM-2 are frozen, and only the transformer encoder and MLP layers are updated during training. The classifier is optimized using a per-token binary cross-entropy loss with logits:

$$\mathcal{L}_{\text{BCE}}(\phi) = -\left[y \cdot \log \sigma(z) + (1 - y) \cdot \log(1 - \sigma(z))\right] \tag{23}$$

where $\sigma(z)$ is the sigmoid activation function and $\mathbf{z} = v_\phi(h)$ is a vector of per-token logit predictions. The loss is computed without reduction to allow for masking padded positions and is averaged over all valid tokens in the batch. v_ϕ is trained on a 1xA6000 NVIDIA DGX server with 50 GB of shared VRAM for 50K steps using the AdamW optimizer (betas=($\beta_1 = 0.99, \beta_2 = 0.98$), weight decay $\lambda = 0.01$), a learning rate (LR) of $3e^{-5}$ with a cosine schedule (5000 warmup steps, LR minimum = $1e^{-5}$). The PET classifier was trained using the same train, test, and validation sequence splits as MemDLM pre-training.

B.4 Computational Metrics

Sequence generation quality was computationally verified using the following metrics:

Pseudo Perplexity The model's generation quality was assessed using the ESM-2 [Lin et al., 2023] pseudo-perplexity metric. Typically, a lower pseudo-perplexity value indicates higher confidence. Specifically, the pseudo-perplexity is computed as the exponential of the negative pseudo-loglikelihood of a sequence. This metric yields a deterministic value for each sequence but necessitates L forward passes for computation, where L represents the input sequence length. It is formally defined as PPL($\mathbf{x}$) = $\exp(-\frac{1}{L}) \sum_{i=1}^{L} \log p(x^i \mid x^{\backslash i})$.

pLDDT The structural confidence of generated sequences was assessed using predicted Local Distance Difference Test (pLDDT) scores from ESMFold v1 with chunk size of 128 [Lin et al., 2023], a protein language model-based tool to predict protein structures from amino acid sequences alone. Higher pLDDT indicates ESMFold is more confident in the produced structure, suggesting the initial input sequence is biologically plausible.

Shannon Entropy To measure the diversity and uncertainty of the model's token predictions, we compute the average Shannon entropy across the sequence. Let $p(x^i)$ denote the model's probability distribution over the vocabulary $\mathcal{V}$ at position i. Higher entropy values indicate greater diversity in the model's predictions, while lower values suggest more repetitive distributions. The entropy is defined as: $\text{Entropy}(\mathbf{x}) = -\frac{1}{L} \sum_{i=1}^{L} \sum_{v \in \mathcal{V}} p(x^i = v) \cdot \log p(x^i = v)$.

BLOSUM62 Substituion Score The average BLOSUM62 score is a quantitative approach to determining whether an amino acid substitution is conservative or nonconservative. This value becomes an important computational metric for protein sequence infilling tasks (both unconditional and PET-based solubilization) to determine if the model is introducing non-conserved residue changes. For each aligned position between a generated sequence $\hat{\mathbf{x}}$ and reference sequence $\mathbf{x}$, we extract the substitution score $B(\hat{x}^i, x^i)$ from the BLOSUM62 matrix [Henikoff and Henikoff, 1992]. Higher scores indicate greater biochemical similarity to the native sequence, while lower scores suggest more divergent or potentially deleterious substitutions. The final score is computed as the mean over all aligned residues $\text{BLOSUM}(\hat{\mathbf{x}}, \mathbf{x}) = \frac{1}{L} \sum_{i=1}^{L} B(\hat{x}^i, x^i)$.

TM Residue Density To estimate the membrane-localizing potential of generated sequences, we used DeepTMHMM v1.0 tool (`https://services.healthtech.dtu.dk/services/DeepTMHMM-1.0/`) [Hallgren et al., 2022] to produce per-residue topology annotations. Each residue is classified into one of six categories: signal peptide (S), inside cell/cytosol (I), alpha membrane (M), beta membrane (B), periplasm (P), or outside cell/lumen (O). For our analysis, we consider residues labeled as alpha membrane (M) to be "soluble" in the membrane context, and all other classes, including beta membrane (B), to be "insoluble." We explicitly exclude B-labeled residues from the soluble category due to the structural and biophysical differences between beta-barrel and alpha-helical transmembrane domains, the latter being dominant in our training set. Using these annotations, we define the *TM Residue Density* of a sequence as the number of residues predicted to lie within alpha membrane ("M" predictions) regions divided by the sequence length as a normalization factor.

B.5 Wet-Lab Experiments

B.5.1 Cloning and Plasmid Construction

DNA sequences of our MemDLM-designed and control peptides were cloned. Target sequences derived from MemDLM were cloned into the pMAL_dstβL vector (Addgene plasmid #73805) between the genes encoding for ToxR and β-lactamase using blunt-end ligation. The resulting constructs were initially transformed into *E. coli* XL-10 Gold cells. Transformants were selected on Luria Broth (LB) agar plates containing spectinomycin and sequences were verified by Sanger sequencing. Confirmed plasmids were subsequently transformed into *E. coli* Cloni cells for the assay.

Cell lines:

REAGENT	CATALOG INFORMATION
E. Cloni 10G DUOs Chemically Competent Cells	Cat. No. 60107-1 (BioSearch Technologies)
XL 10-Gold Ultracompetent Cells	Cat. No. 200315 (Agilent)

Table 4: Competent cell reagents used in this study.

Genes inserted into the pMAL_dstβL plasmid vector:

- **Human CLS:**
  - Uniprot: UPI000007083D
  - Amino acid sequence: PLFIPVAVMVTAFSGLAFIIWLA
  - Gene: CCGCTGTTCATCCCGGTTGCAGTTATGGTTACCGCTTTTAGTGGATTG-GCGTTTATCATCTGGCTGGCT
- **GpA-TM Region:**
  - Uniprot: UPI000012B75E
  - Amino acid sequence: LIIFGVMAGVIGTILI
  - Gene: TTAATTATTTTCGGAGTGATGGCCGGAGTTATCGGCACAATTTTAATC
- **ErbB2 TM Region:**
  - Uniprot: P04626-1
  - Amino acid sequence: SIISAVVGILLVVVLGVVFGIL
  - Gene: TCCATTATCTCCGCTGTCGTAGGAATCTTGTTAGTTGTCGTC-CTTGGGGTTGTGTTTGGAATTTTA
- **Qsox2 TM Region:**
  - Uniprot: Q6ZRP7
  - Amino acid sequence: SLCVVLYVASSLFMVMYFF

  - Gene: AGTCTTTGCGTCGTACTTTACGTCGCATCTTCACTGTTTATGGTGATGTATTTCTTT

- **EK3 Water Soluble Helix** [Wolny et al., 2017]:
  - Amino acid sequence: SAEEEKKKAEEEKKKAEEEKKKAE
  - Gene: TCCGCAGAGGAAGAAAAGAAAAAAGCTGAAGAAGAAAAGAAAAAGGCAGAAGAAGAGAAAAAAAAGGCAGAG
- **PoorTM2**
  - MemDLM amino acid sequence: SSLLFSYQGAKKEEERVFLDNF
  - Gene: AGTTCTTTGTTATTCAGCTATCAGGGAGCCAAGAAAGAAGAAGAACGTGTGTTTCTGGATAACTTC
- **PoorTM4**
  - MemDLM amino acid sequence: GTHAKDWRVTSWKRYGEIE
  - Gene: GGAACACATGCTAAAGATTGGCGTGTGACATCTTGGAAGCGTTACGGCGAGATTGAA
- **GoodTM4**
  - MemDLM amino acid sequence: DLSKWLGIVLLLLLAILALLLIR
  - Gene: GATTTAAGCAAATGGCTGGGTATCGTACTGTTACTGTTACTGGCTATTTTGGCTTTATTACTGATTCGT
- **GoodTM5**
  - MemDLM amino acid sequence: SLRWLWSLVIGLLLIVAFYLLLR
  - Gene: AGCCTGCGTTGGTTGTGGTCTTTAGTGATCGGCTTACTGCTTATCGTTGCCTTCTACCTGCTGCTTCGC
- **GoodTM8**
  - MemDLM amino acid sequence: DFLRKAVIVLLVLVIVAGLLVIR
  - Gene: GATTTTCTGCGTAAGGCAGTGATTGTATTACTTGTCTTGGTTATTGTGGCGGGTCTGCTGGTTATTCGC

B.5.2 TOXCAT-β-Lactamase Growth Assay

Single colonies of plasmid-containing E. *coli* Cloni cells were used to inoculate 6-mL LB cultures supplemented with 50 μg/mL spectinomycin. Glycerol stocks were made and used to inoculate new fresh LB culture tubes with 50 μg/mL spectinomycin. Cultures were incubated for $\sim$8 h or overnight at 37$\circ$C with shaking. Optical density at 600 nm (OD_{600}) was measured, and cultures were diluted with fresh LB + spectinomycin to an OD_{600} of 0.05. Growth was continued until an OD_{600} of $\sim$0.1 was reached.

To ensure consistent inoculation density across assays, the number of cells per well was normalized to 1.95×10^5 cells. This value was calculated using the relationship of 1 $OD_{600} \approx 8 \times 10^8$ cells/mL and adjusted for the measured absorbance at OD_{600} of each culture. Growth under spectinomycin confirmed that the pMal_dsTBL plasmid was successfully introduced into E. *coli* Cloni cells across all conditions. All cultures grew equally under this condition, demonstrating comparable inoculation densities and consistent plasmid uptake.

Assays were performed in 96-well plates, with each well containing a final total volume of $\sim$200 μL LB medium supplemented with the appropriate antibiotics in the following concentrations: Spectinomycin (50 μg/mL), Carbenicillin (300 μg/mL), Carbenicillin (100 μg/mL) + chloramphenicol (100 μg/mL), Carbenicillin (100 μg/mL) + chloramphenicol (120 μg/mL). Wells were inoculated with the calculated volume of diluted culture corresponding to 1.95×10^5 cells. Each antibiotic reporter was run in triplicate. Plates were incubated at 37°C in a pre-heated plate reader (BioTek Synergy H1). Bacterial growth was monitored by measuring absorbance at 600 nm for 24 hours with measurements taken every 10 minutes under continuous shaking.

C Extended Results

C.1 Density Plots

We visualize the density distribution of the various computational metrics to assess membrane protein sequences. When using P2 Self-Planning to generate sequences, we set $\tau = 0.7$ to have a slight bias towards deterministic model outputs.

Unconditional Generation We unconditionally generate 1,000 membrane protein sequences. Lengths are randomly chosen from 50-250 residues.

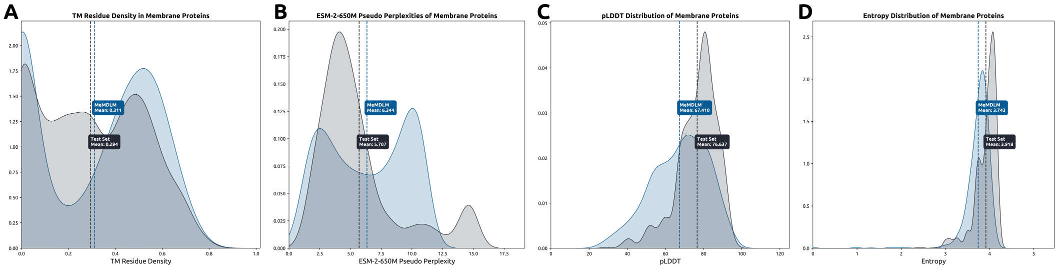


Figure A1: *De novo*-generated and natural membrane protein sequences.

Motif Scaffolding We mask out and infill both the insoluble and soluble regions of natural membrane proteins derived from the model's test set.

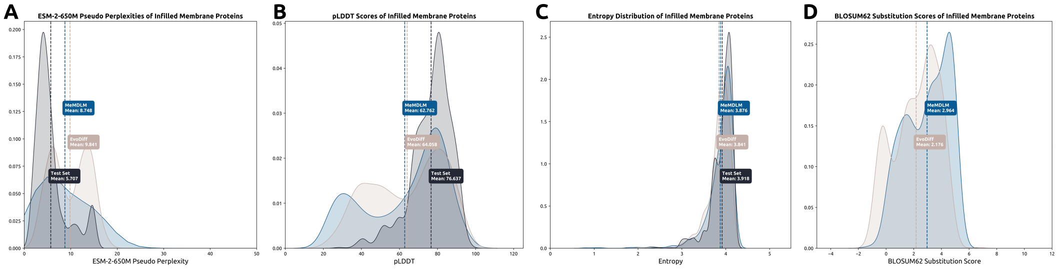


Figure A2: Infilling Insoluble Domain

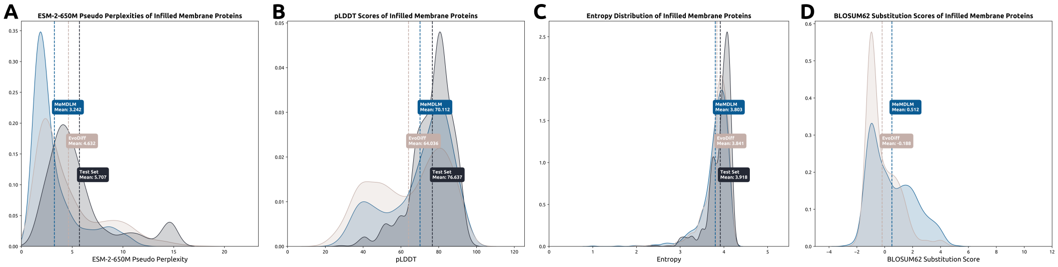


Figure A3: Infilling Soluble Domain

Solubilization We optimize the solubility of the proteins in the model's test set by applying our PET algorithm.

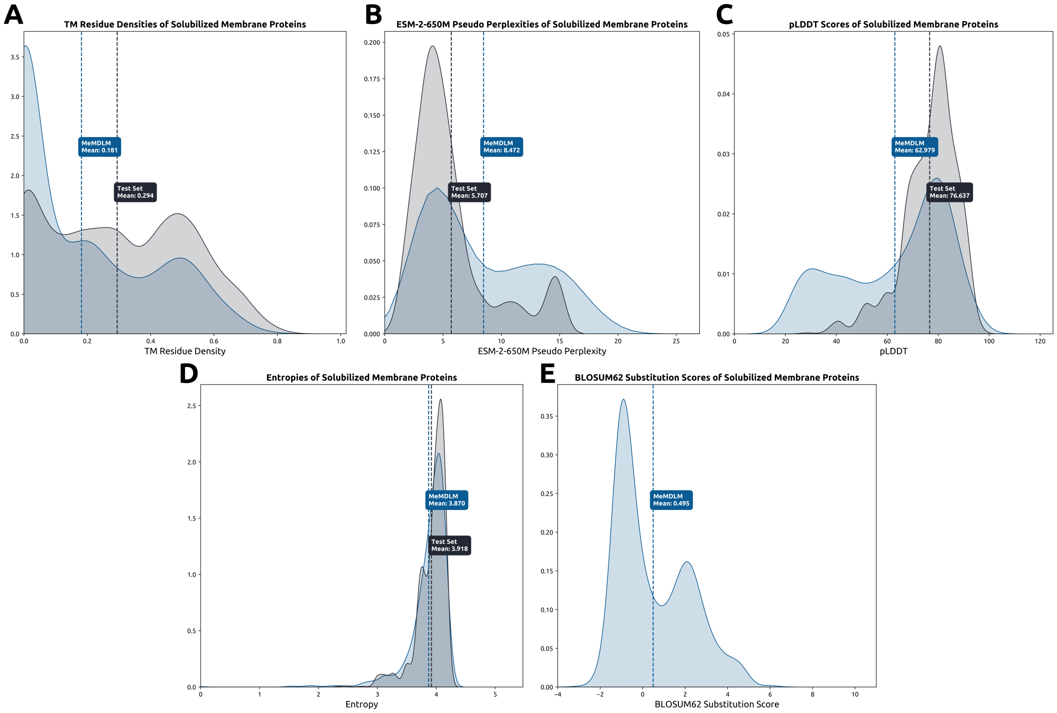


Figure A4: Solubilizing TM Domains

C.2 Physciochemical Property Prediction

As a surrogate task, we assessed if RDM training retains physicochemical information critical to membrane protein function by predicting per-residue solubility and membrane localization (Table 3). We use embeddings from three models–vanilla ESM-2-650M, ESM-2-650M fine-tuned on membrane protein sequences, and MemDLM–as inputs to a per-residue solubility and sequence-level membrane localization classifiers. We outline the dataset, training details, and evaluation results of these models in the following.

C.2.1 Datasests

Solubility Prediction We leveraged the same set of 11,908 membrane sequences from the MemDLM training dataset to develop a binary classifier that predicts the solubility of each amino acid within a protein sequence. Each sequence was annotated on a per-residue basis, with TM (class 1) and soluble (class 0) labels assigned according to the sequence's uppercase and lowercase residues, respectively. The same training, testing, and validation data splits used to train MemDLM were also utilized to train and evaluate this classifier.

Membrane Localization We collected 30,020 protein sequences from DeepLoc 2.0 thumuluri2022deeploc to build a binary classifier that predicts a protein sequence's cellular localization. The authors of the dataset provided a multi-label label for each sequence indicating its localization(s). We used the authors' provided data splits, with training sequences having 11 labels and testing sequences having 8 labels.

C.2.2 Models

Solubility Prediction We first predicted TM and soluble residues, a hallmark characteristic of membrane protein sequences. We utilized embeddings from each pLM's latent space (ESM-2-150M, ESM-MLM, and MemDLM) as inputs to train a two-layer perceptron classifier that minimized the standard binary cross-entropy (BCE) loss to compute the probability that each residue in the sequence is either soluble (probability < 0.5, class 0) or TM (probability > 0.5, class 1).

Membrane Localization Prediction Proteins originating from the endomembrane system and localizing in the plasma membrane differ in conformation and function from those in the cytosol and other cellular organelles. We predicted the subcellular localization of protein sequences by utilizing embeddings from each pLM's latent space (ESM-2-150M, ESM-MLM, and MemDLM) to train a XGBoost classifier that minimized the standard BCE loss to compute the probability that a protein sequence localizes in the plasma membrane (probability > 0.5, class 1) or in other regions (probability < 0.5, class 0).

Fine-Tuning ESM-2 We fine-tune the ESM-2 pLM ([Lin et al., 2023]) to achieve an encoder that produces membrane-aware protein sequence embedding used as a baseline comparison for the RDM training task. We trained a MLM head on top of ESM-2-650M using membrane protein sequences to force comprehension of membrane protein properties. We chose to randomly mask 40% of amino acid tokens during training over the standard 15% to more closely resemble the dynamics of diffusion-based (RDM) training; masking rates above 40% have been seen as detrimental during MLM training tasks [Wettig et al.]. Corrupted sequences were passed into ESM-2-650M to retrieve their output embeddings. During training, we unfroze the key, query, and value weights in the attention heads of the final three encoder layers, similar to fine-tuning EvoFlow during MemDLM training. During ESM-2 fine-tuning, the model performed a *masked-prediction* task over masked amino acid tokens to minimize the NLL loss in Eq. (15). 2xH100 NVIDIA GPUs, learning rate of 5e-3, the Adam optimizer, and a batch size of 8 over 10 epochs were used.

C.2.3 Results

We leveraged the trained solubility prediction and membrane localization classifiers to determine if latent spaces from RDM-based generative models are aligned with relevant membrane protein properties. Table 5 shows that MemDLM latent embeddings achieve predictive performance that closely parallels SOTA pLM embeddings, which are designed specifically for delivering precise representations.

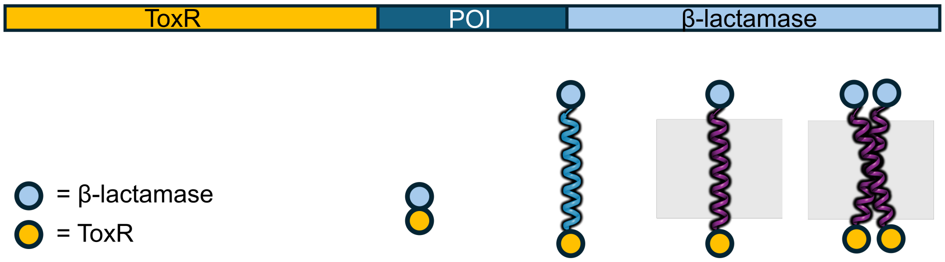
MODEL	SOLUBILITY (↑)	MEMBRANE LOCALIZATION (↑)
ESM-2-650M	0.9383	0.6011
Fine-Tuned ESM-2	0.9375	0.6000
MemDLM	0.9375	0.5964

Table 5: Performance comparison (AUROC) of embeddings derived from various models in predicting physico-chemical properties of MemDLM test set sequences.

In total, these results demonstrate that MemDLM accurately captures the biological features underpinning functional membrane proteins despite being trained on a sequence generation task rather than a masked-prediction task.

C.3 Wet-Lab Experiments

C.3.1 TOXCAT Assay



	ΔToxR-POI-βL	ΔPOI	Soluble	TM (monomer)	TM (dimer)
Control Samples	ΔToxR-POI-βL	ΔPOI	EK3	CLS	ErbB2, GpA, Qsox2
Spectinomycin	Growth	Growth	Growth	Growth	Growth
Carbenicillin	No growth	No growth	No growth	Growth	Growth
Carbenicillin + Chloramphenicol	No growth	No growth	No growth	No growth	Growth

Figure A5: Summary of control constructs for the TOXCAT-β-lactamase assay and their expected growth responses to antibiotics.

Schematic showing gene ToxR-POI-βL, where POI is the peptide of interest and βL is β-lactamase. Periplasmic β-lactamase and cytoplasmic ToxR proteins are represented by blue and yellow dots, respectively. Expected growth phenotypes under spectinomycin and carbenicillin +/-chloramphenicol are indicated for each control. Negative controls ΔToxR-POI-βL, ΔPOI, and EK3 should not survive in carbenicillin because they lack a TM domain. Positive controls CLS, ErbB2, GpA, and Qsox2 all have TM domains and should survive in carbenicillin. Further, ErbB2, GpA, and Qsox2 are dimers. Expression of these controls should also confer resistance to chloramphenicol.

C.3.2 TOXCAT Sequence Selection

From 1,000 MemDLM-generated sequences, three sequences from the top 100 predicted performers ("GoodTM") and two sequences from the bottom 22 predicted performers ("PoorTM") were selected for screening in the TOXCAT assay. The following selection criteria was used:

CATEGORY	pLDDT	PPL	TM RESIDUE DENSITY	SEQUENCES SELECTED
GoodTM (Top 100)	> 60	< 10	Non-zero	3
PoorTM (Bottom 22)	< 60	< 15	Non-zero	2

Table 6: Selection criteria and sequence counts for MemDLM-generated sequences screened in the TOXCAT assay.

The top-ranked (GoodTM) sequences represented a diverse set of high-scoring designs. For example, GoodTM5 (SLRWLWSLVIGLLLIVAFYLLLR, rank 57) contained a small-X_3-small motif known to promote TM helix association [Russ and Engelman, 1999] [Li et al., 2004] [Russ and Engelman, 2000]. This further demonstrates that MemDLM generates plausible protein sequences with TM-like character.

C.3.3 Growth Curves

Control Plasmids Growth curves of *E. coli* Cloni cells containing control plasmids.

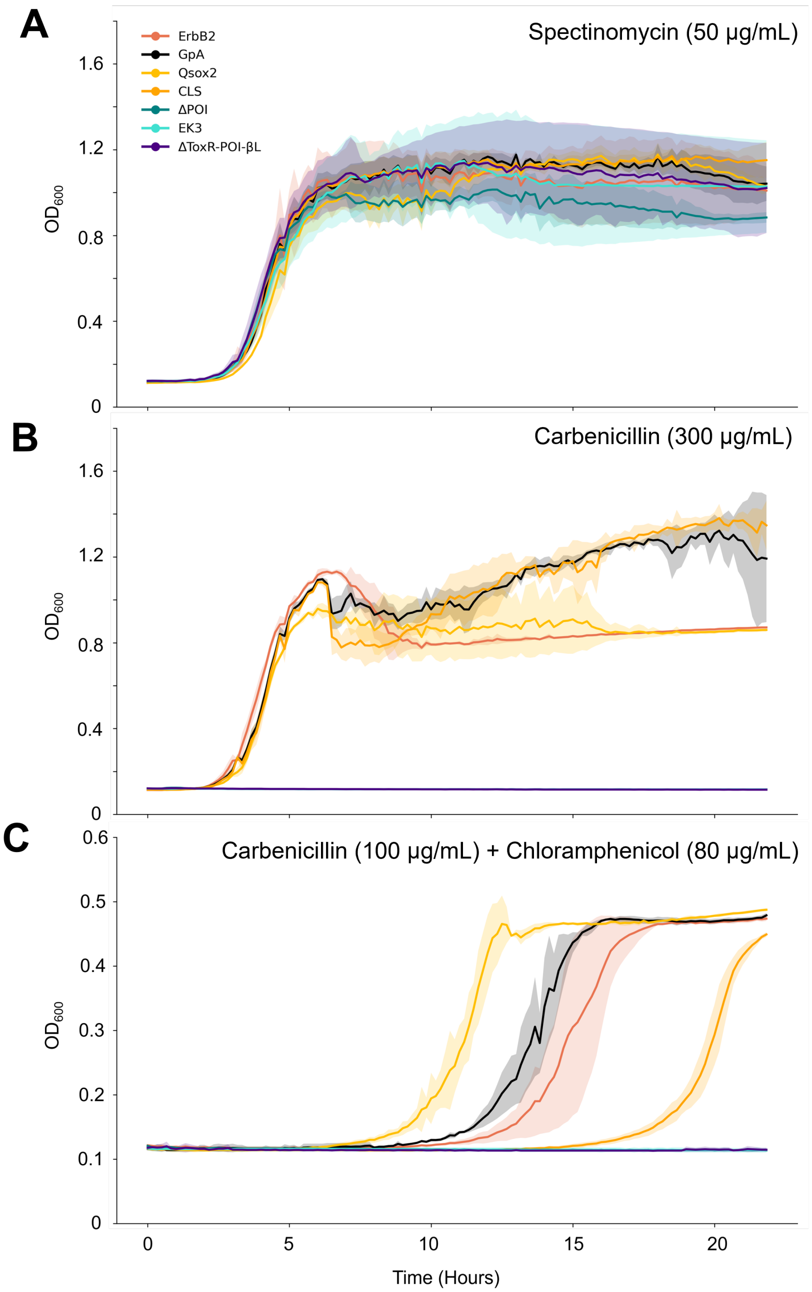


Figure A6: **A)** Survival in spectinomycin (50 μg/mL) confirmed plasmid uptake for all controls. **B)** Growth curves of control plasmids under carbenicillin (300 μg/mL) showed that control plasmids containing TM sequences survived selective pressure. **C)** Growth curves of control plasmids under combined carbenicillin (100 μg/mL) and chloramphenicol (80 μg/mL) selection, which tests both transmembrane insertion and association, show that the dimeric Qsox2, GpA, and ErbB2 controls begin growing in chloramphenicol earlier than the monomeric CLS control.

MemDLM-Generated Sequences Growth curves for MemDLM's *de novo*-generated sequences.

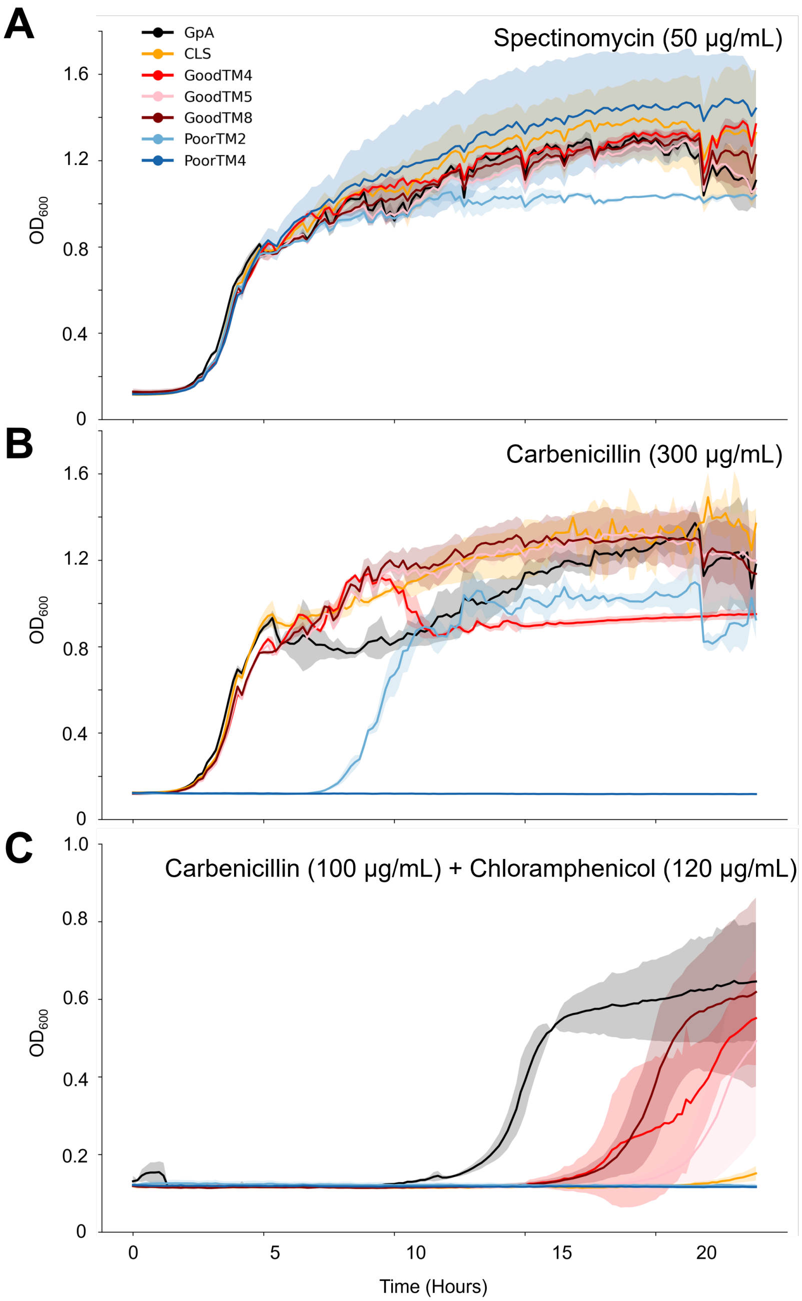


Figure A7: GpA is used as a positive control for insertion and TM association. CLS is the positive insertion and negative TM association control. **A)** Growth curve of *E. coli* Cloni cells containing *de novo* MemDLM TM sequences under spectinomycin (50 μg/mL) confirmed plasmid uptake. **B)** Growth curves of MemDLM peptides under carbenicillin (300 μg/mL) show GoodTM4, GoodTM5, and GoodTM8 growing as expected. PoorTM4 did not survive, indicating that it is not membrane inserting. PoorTM2 showed delayed growth, suggesting that it has lower membrane insertion propensity than the GoodTM constructs. **C)** Growth curves of MemDLM plasmids under combined carbenicillin (100 μg/mL) and chloramphenicol (120 μg/mL), used to select for both transmembrane insertion and transmembrane association, reveal that some of the TM designs may be oligomeric.

D Visualizations

AlphaFold3 visualizations of MemDLM-generated membrane protein sequences. TM Residue Density (TMRD) scores are derived from DeepTMHMM predictions. Structures and colors are from AlphaFold3 predictions, and pLDDT scores are from ESMFold.

D.1 *De novo* Generation

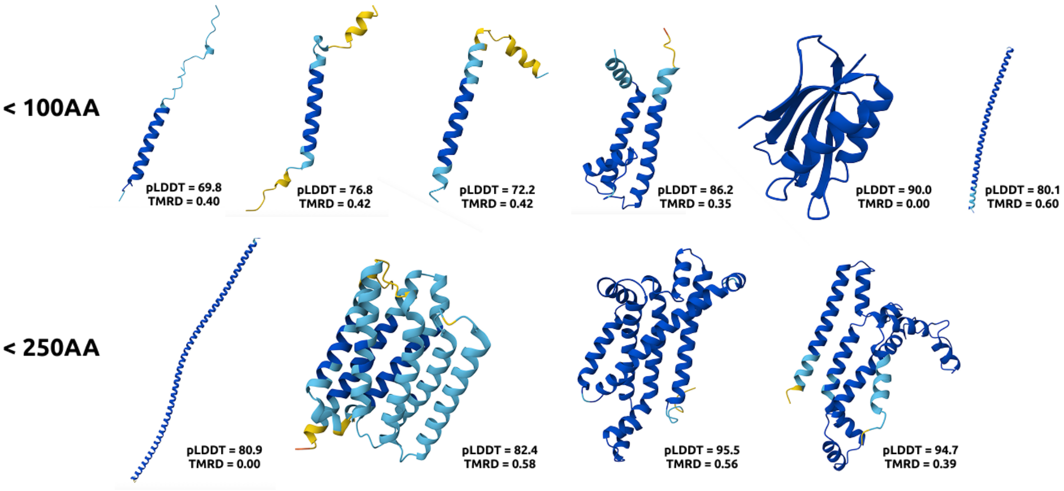
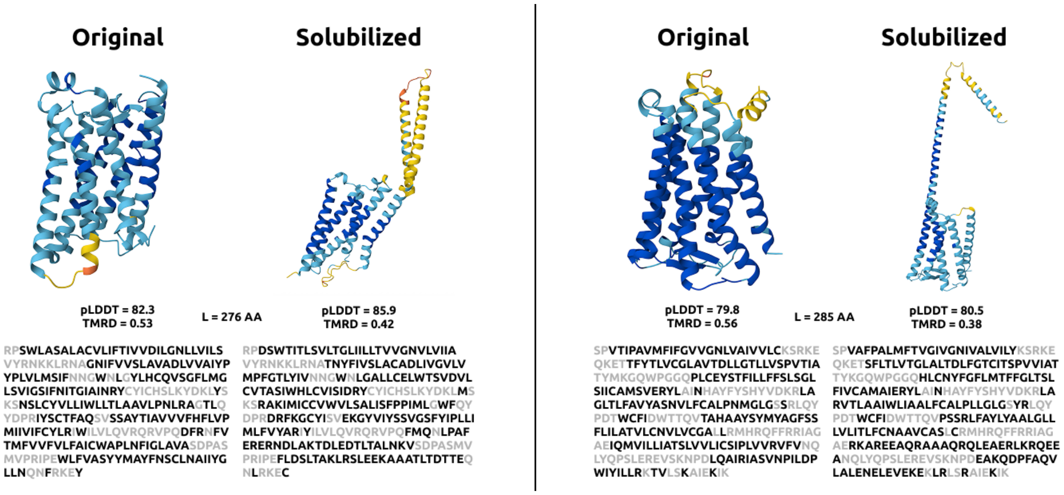


Figure A1: *De novo*-generated protein sequences from MemDLM across different lengths.

D.2 Solubilization



Original
Solubilized
pLDDT = 82.3
TMRD = 0.53
L = 276 AA
pLDDT = 85.9
TMRD = 0.42
RPSWLASALACVLIFTIVVDILGNLLVILS
VYRNKKLRNAGNIFVVSLAVADLVVAIYP
YPLVLMSIFNNGWNLGYLHCQVSGFLMG
LSVIGSIFNITGIAINRYCYICHSLKYDKLYS
KSNSLCYVLLIWLLTLAAVLPNLRAGTLQ
YDPRIYSCTFAQSVSSAYTIAVVVFHFLVP
MIIVIFCYLRIWILVLQVRQRVPQDFRNFV
TMFVVFVLFAICWAPLNFIGLAVASDPAS
MVPRIPEWLFVASYYMAYFNSCLNAIIYG
LLNQNFRKEY
RPDSWTITLSVLTGLIILLTVVGNVLVIIA
VYRNKKLRNATNYFIVSLACADLIVGVLV
MPFGTLYIVNNGWNLGALLCELWTSVDVL
CVTASIWHLCVISIDRYCYICHSLKYDKLMS
KSRAKIMICCVWVLSALISFPPIMLGWFQY
DPRDRFKGCYISVEKGYVIYSSVGSFYIPLLI
MLFVYARIYILVLQVRQRVPQFMQNLPAF
ERERNDLAKTDLEDTLTALNKVSDPASMV
PRIPEFLDSLTAKLRSLEEKAAATLTDTTEQ
NLRKEC
Original
Solubilized
pLDDT = 79.8
TMRD = 0.56
L = 285 AA
pLDDT = 80.5
TMRD = 0.38
SPVTIPAVMFIFGVVGNLVAIVVLCKSRKE
QKETTFYTLVCGLAVTDLLGTLLVSPVTIA
TYMKGQWPGGQPLCEYSTFILLFFSLSGL
SIICAMSVERYLAINHAYFYSHYVDKRLA
GLTLFAVYASNVLFCALPNMGLGSSRLQY
PDTWCFIDWTTQVTAHAAYSYMYAGFSS
FLILATVLCNVLVCGALLRMHRQFFRRIAG
AEIQMVILLIATSLVVLICSIPLVVRVFVNQ
LYQPSLEREVSKNPDLQAIRIASVNPILDP
WIYILLRKTVLSKAIEKIK
SPVAFPALMFTVGIVGNIVALVILYKSRKE
QKETSFLTLVTGLALTDLFGTCITSPVVIAT
TYKGQWPGGQHLCNYFGFLMTFFGLTSL
FIVCAMAIERYLAINHAYFYSHYVDKRLA
RVTLAAIWLIAALFCALPLLGLGSYRLQY
PDTWCFIDWTTQVPSSRLFAYLYAALGLL
LVLITLFCNAAVCASLCRMHRQFFRRIAG
AERKAREEAQRAAAQRQLEAERLKRQEE
ANQLYQPSLEREVSKNPDEAKQDPFAQV
LALENELEVEKEKLRLSRAIEKIK

Figure A2: Original and solubilized versions of MemDLM test set protein sequences. Grey residues were annotated as soluble in the given sequence and were thus "fixed" during PET sampling.

E Algorithm Pseudocode

Algorithm 1 MemDLM Training

```
Require: Protein sequence dataset $\mathcal{D}$, diffusion model $p_\theta$, 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 $p_\theta$
```

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 $p_\theta$, 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 $\mathbf{x}_0$
```

Algorithm 3 PET-based MemDLM Sampling

```
Require: Candidate protein sequence $\mathbf{x}$, trained MemDLM $p_\theta$, trained solubility classifier $v_\phi$, pre-trained encoder Encoder$_\phi$, 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 $\hat{x}^i \sim \textsc{Cat}(\log P(x^i))$
    Update $\mathbf{x}[i] \leftarrow \hat{x}^i$
end for
return Optimized sequence $\hat{\mathbf{x}}$
```