MEMDLM: *De Novo* MEMBRANE PROTEIN DESIGN WITH PROPERTY-GUIDED DISCRETE DIFFUSION

Shrey Goel,[1] **Vishrut Thoutam,**[2] **Edgar Mariano Marroquin,**[3]**, Aaron Gokaslan,**[3]
Arash Firouzbakht,[4] **Sophia Vincoff,**[1]**, Volodymyr Kuleshov,**[3] **Huong T. Kratochvil,**[4]
Pranam Chatterjee[1,†]

[1]Duke University, Durham, NC
[2]High Technology High School, Lincroft, NJ
[3]Cornell Tech, New York, NY
[4]University of North Carolina, Chapel Hill, NC

[†]Corresponding author: pranam.chatterjee@duke.edu

ABSTRACT

Masked Diffusion Language Models (MDLMs) have recently emerged as a strong class of generative models, paralleling state-of-the-art (SOTA) autoregressive (AR) performance across natural language modeling domains. While there have been advances in AR as well as both latent and discrete diffusion-based approaches for protein sequence design, masked diffusion language modeling with protein language models (pLMs) is unexplored. In this work, we introduce **MeMDLM**, an MDLM tailored for membrane protein design, harnessing the SOTA pLM ESM-2 to *de novo* generate realistic membrane proteins for downstream experimental applications. Our evaluations demonstrate that MeMDLM-generated proteins exceed AR-based methods by generating sequences with greater transmembrane (TM) character. We further apply our design framework to scaffold soluble and TM motifs in sequences, demonstrating that MeMDLM-reconstructed sequences achieve greater biological similarity to their original counterparts compared to SOTA inpainting methods. Finally, we apply a generalized Bayesian optimization procedure that uniquely uses saliency maps to facilitate the generation of soluble membrane proteins, paving the way for experimental applications. In total, our pipeline motivates future exploration of MDLM-based pLMs for protein design.

1 MEANINGFULNESS STATEMENT

Meaningful representations of life capture the complexity and function of biological systems in a way that enables the engineering of life's building blocks. We contribute to this direction by introducing MeMDLM, an MDLM that not only generates realistic membrane protein sequences but also retains strong numerical representations of important membrane protein properties. MeMDLM advances protein design efforts by eliminating the need to learn from protein structure, a particularly challenging prerequisite for membrane proteins. Overall, our work enables the design of novel proteins with potential downstream experimental and therapeutic applications.

2 INTRODUCTION

Membrane proteins are essential for molecular transport, signal transduction, and cellular communication, making them critical therapeutic targets (Jelokhani-Niaraki, 2022; Sanganna Gari et al., 2021). However, *de novo* design of membrane proteins is challenging due to reliance on structure-based models and limited high-resolution structural data, with only 1% of PDB entries representing membrane proteins (Wang et al., 2022; Yin et al., 2007; Elazar et al., 2022; Vorobieva et al., 2021). Sequence-based models offer an alternative, but autoregressive (AR) approaches struggle with protein design tasks because they generate residues sequentially, limiting long-range dependency modeling—a critical feature for the complex topology of membrane proteins (Ferruz et al., 2022). To address these limitations, we introduce MeMDLM, a classifier-guided masked diffusion language model (MDLM) that enables parallel, non-sequential generation, capturing global sequence dependencies without structural input.

MeMDLM leverages the MDLM framework to generate novel membrane protein sequences by iteratively masking and reconstructing amino acid tokens, allowing the model to learn relationships across distant residues (Sahoo et al., 2024). This contrasts with AR models, which are biased toward local context and can miss critical global interactions necessary for membrane protein stability and function. We further integrate LaMBO-2 for classifier-guided sampling (Gruver et al., 2024), optimizing sequences for solubility and transmembrane (TM) characteristics. Our results show that MeMDLM generates sequences with TM residue distributions closely matching natural membrane proteins, outperforms AR models in capturing transmembrane features, and successfully scaffolds functional motifs. These advances highlight MDLM's unique ability to model the global sequence constraints essential for membrane protein design.

3 Methods

3.1 Masked Diffusion Language Model (MDLM)

MDLM is a discrete diffusion architecture that retrains MLMs to learn the true distribution of data by reconstructing sequences noised with `<MASK>` tokens. The MDLM training task leverages the absorbing-state forward diffusion process and a specific reverse diffusion parameterization to simplify the loss function and increase model accuracy. The absorbing state diffusion process, $q(\mathbf{z_t}, \mathbf{x})$ is a distribution parameterized by a time-conditioned log-linear noise schedule $\alpha_t = -\log(1-t)$. During training, we sample timesteps $t \sim \mathcal{U}(0,1)$ to compute α_t, the probability of clean data $\mathbf{x_0}$ remaining unchanged, and $1-\alpha_t$, the probability of $\mathbf{x_0}$ transitioning to a `<MASK>` token:

$$q(\mathbf{z_t}, \mathbf{x_0}) = \text{Cat}(\mathbf{z_t}; \alpha_t \mathbf{x_0} + (1-\alpha_t)\mathbf{m}) \tag{1}$$

The reverse diffusion process, matching the estimated forward diffusion posterior $p(\mathbf{z_s} \mid \mathbf{z_t})$, is parameterized by a categorical distribution ("SUBS") that enforces restrictions on the original discrete diffusion formulation specific to absorbing state diffusion methods. During the SUBS-parameterized reverse process, unmasked tokens are unchanged and masked tokens are guaranteed to be unmasked:

$$p_\theta(\mathbf{z_s}|\mathbf{z_t}, \mathbf{x}) = \begin{cases} \text{Cat}(\mathbf{z_s}; \mathbf{z_t}) & \mathbf{z_t} \neq \mathbf{m} \\ \text{Cat}\left(\mathbf{z_s}; \dfrac{(1-a_s)\mathbf{m} + (a_s - a_t)\mathbf{x}}{1-a_t}\right) & \mathbf{z_t} = \mathbf{m} \end{cases} \tag{2}$$

We utilize the ESM-2-150M pLM as the backbone model for learning the denoising network $x_\theta(\mathbf{z_t})$ that reconstructs the original sequence from its masked counterpart (Lin et al., 2023). Because SUBS "carries-over" unmasked tokens and masking rates are scheduled in a log-linear fashion, batches with 100% masking rates are problematic because x_θ does not have contextual information to guide the denoising process. Thus, we employ a maximal masking rate $\alpha_{\text{max}} = 0.75$ to ensure our denoising network learns long-range sequence dependencies while still training on higher masking rates to facilitate *de novo* generation.

With SUBS parameterization, we minimize a modified continuous-time NELBO, a Rao-Blackwellized form of the original D3PM loss (Ho et al., 2020) that eliminates the reconstruction loss term:

$$\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_{q,t}\left[-\log p_\theta(\mathbf{x}|\mathbf{z}_{t(0)}) + T\left[\frac{a_t - a_s}{1-a_t}\log\langle x_\theta(\mathbf{z_t}), \mathbf{x}\rangle\right]\right] \tag{3}$$

Overall, MeMDLM is a fine-tuned encoder that unconditionally generates membrane-like protein sequences and produces membrane-aware embedding (Figure 1). To enable the ESM-2 pLM with principled generation capabilities, we first pre-train ESM-2-150M on the MDLM task using sequences that span the entire protein space, then fine-tune this model with membrane proteins to facilitate *de novo* membrane protein sequence generation.

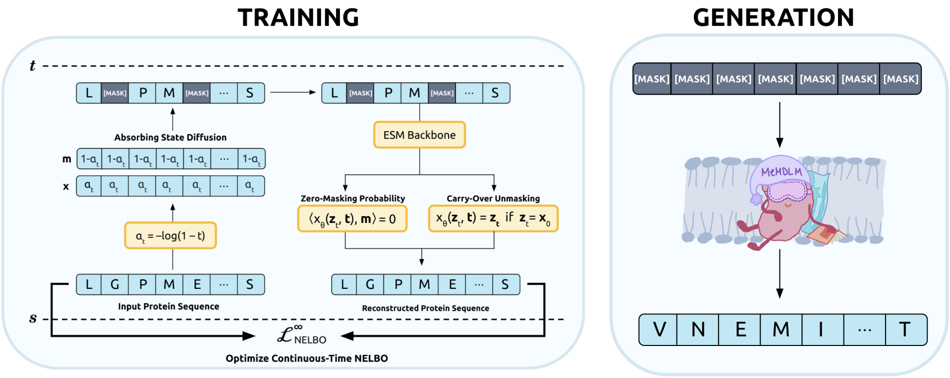
TRAINING
GENERATION
t
L [MASK] P M [MASK] ··· S
Absorbing State Diffusion
m 1-a_t 1-a_t 1-a_t 1-a_t 1-a_t ··· 1-a_t
x a_t a_t a_t a_t a_t ··· a_t
a_t = –log(1 – t)
L G P M E ··· S
Input Protein Sequence
ESM Backbone
Zero-Masking Probability
⟨x_θ(z_t, t), m⟩ = 0
Carry-Over Unmasking
x_θ(z_t, t) = z_t if z_t = x_0
Reconstructed Protein Sequence
s
ℒ^∞_NELBO
Optimize Continuous-Time NELBO
[MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK]
MeMDLM
V N E M I ··· T

Figure 1: Denoising and noising processes guided by SUBS parameterization in MeMDLM. Protein sequences are corrupted according to the noising scheduler a_t and denoised via x_θ (ESM-2), calculating loss between the true and reconstructed sequence.

3.2 Classifier-Guided Masked Discrete Diffusion

Preliminaries LaMBO-2 is a powerful Bayesian optimization algorithm that enables multi-objective protein design via discrete classifier guidance Gruver et al. (2024). It extends the popular continuous guidance strategy that biases the diffusion trajectory toward a target class y using the gradients $\nabla_{x_t} \log \ v(y \mid x_t, t)$, where $v(y \mid x_t, t)$ is an external classifier trained on noisy data (Dhariwal & Nichol, 2021). Specifically, LaMBO-2 follows the gradients of v_θ, a value function trained on corrupted sequences and hidden states e derived from the learned diffusion model x_θ, to circumvent the lack of continuous representations in discrete gradient guidance. We extend LaMBO-2 to the MDLM sampling process to introduce token-level optimization over the sequence-level optimization originally employed by LaMBO-2. We present the full sampling algorithms in Supplementary G.

Token-level Guidance We propose directly updating the hidden states e_0 of a seed sequence w_0 to avoid taking discrete jumps in the logits matrix. This approach naturally compliments the design of MeMDLM, which uses its continuous representations as inputs to its language modeling head x_θ^L that generates logits. To optimize continuous sequence representations, we introduce parameters Δ_{saliency} and Δ that aggregate saliency information and an explore-exploit loss, respectively.

Implementation We train v_θ to assign per-residue solubility scores to an unoptimized (insoluble) seed sequence that was unconditionally generated from x_θ. v_θ's training sequences are corrupted according to the transition probability categorical distribution from equation 1; however, we set $\alpha_{\text{max}} = 0.50$ to approximately match average soluble residue density of the test set. Our focus on per-residue scores preserves each residue's contribution to the overall sequence score, which is an important basis for token-level guidance. Using v_θ, we follow LaMBO-2's selection of edit positions that do not contribute to the overall guidance objective by constructing the saliency map $\mathbf{S} \in \mathbb{R}^l$ for a sequence of length l, where $\mathbf{S}_i$ represents the saliency at the token $i \in \{0, 1, \ldots, l\}$:

$$\mathbf{S}_i = \max\left\{ \left(\sum_{j=1} \left| \nabla_{\Delta_{\text{saliency}}} v_\theta(e')_{ij} \right| \right)^{1/\tau}, \epsilon \right\}, \qquad p_{\text{edit}}(w) = \frac{\mathbf{S}_i}{\sum \mathbf{S}_i} \tag{4}$$

using the parameters $\tau = 0.9$ and $\epsilon = e^{-4}$. With this selection procedure, high values of $\mathbf{S}_i$ correlate to tokens that do not contribute to the overall sequence value and thus concentrate the largest edit probability mass. We further down-select edit positions with top-k sampling:

$$K_{\text{edit}} = \text{top-k}(p_{\text{edit}}(w), k = \min\{10, p_{\text{edit}}(w) \neq 0\}) \tag{5}$$

We utilize the indices of K_{edit} to create $\mathbf{m} \in \mathbb{R}^l$, a one-hot vector encoding select edit positions in the sequence, enabling us to zero the gradient contribution for high-value (soluble) tokens with the step $h' \leftarrow x_\theta^L(e' + \Delta_{\text{saliency}} \odot \mathbf{m})$. We apply an explore-exploit loss to the logits:

$$\mathcal{L}_{\text{EE}} = \lambda \left[D_{\text{KL}}(h' \| h) - \sum_{i=1} v_\theta(e')_i \right], \quad h' \leftarrow h' + x_\theta^L(e' + \nabla_\Delta \mathcal{L}_{\text{EE}}) \tag{6}$$

where gradients $\sum_{i=1} v_\theta(e')_i$ encourage high values of the desired sequence property while the KL term ensures the transition distribution maximizes the original likelihood. We chose to omit the stochasticity term $\sqrt{2\eta\tau}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0,1)$ from LaMBO-2's Langevin dynamics update step to focus on deterministic updates that will optimize the current sequence, rather than exploring alternative sequences that may not align with the optimization objective.

Sampling During inference, we first generate an unconditional seed sequence with hidden states e_0 using N MeMDLM sampling steps. We refine this sequence with classifier guidance, performing N additional sampling steps and applying S optimization steps at each N_i-th step. We initialize the prior as $h_0 = x_\theta^L(e_0)$ and update it at each N_i-th step with equation 6 to ensure logits integrate both saliency-guided and explore-exploit-driven updates. Our approach (Figure 2) unifies classifier guidance with MeMDLM, enabling controllable discrete diffusion in a continuous latent space.

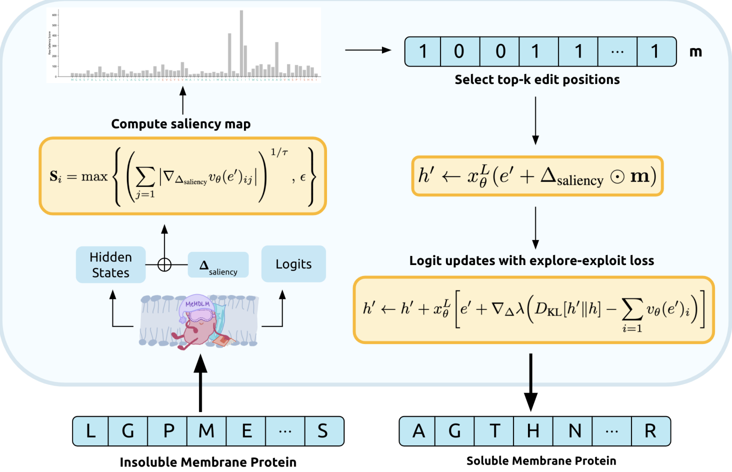
Compute saliency map
$\mathbf{S}_i = \max\left\{ \left(\sum_{j=1} |\nabla_{\Delta_{\text{saliency}}} v_\theta(e')_{ij}| \right)^{1/\tau}, \epsilon \right\}$
Hidden States
Δ_{saliency}
Logits
MeMDLM
1 0 0 1 1 ··· 1
m
Select top-k edit positions
$h' \leftarrow x_\theta^L(e' + \Delta_{\text{saliency}} \odot \mathbf{m})$
Logit updates with explore-exploit loss
$h' \leftarrow h' + x_\theta^L\left[e' + \nabla_\Delta \lambda\left(D_{\text{KL}}[h' \| h] - \sum_{i=1} v_\theta(e')_i\right)\right]$
L G P M E ··· S
Insoluble Membrane Protein
A G T H N ··· R
Soluble Membrane Protein

Figure 2: Refining continuous sequence representation by following gradients of the value function, culminating in token-level optimization for property-guided protein design.

4 Results

To identify the effects of choosing saliency-based edit positions, we visualized saliency scores computed by equation 4 for randomly selected sequences in the MeMDLM test set. In our framework, tokens with higher saliency scores are more likely to be selected for optimization. Figure 3 demonstrates that insoluble (teal) regions have high saliency scores while soluble (orange) regions have low saliency scores after completing S optimization steps after only the N_0-th generation step.

The results suggest that logit updates in equation 6 are deterministic, driven by the gradients of the classifier rather than stochastic noise. By drawing the connection between saliency scores and discrete token optimization, we show that guiding sequence design in a discrete space is informed by the continuous, latent structure of the sequence.

Given the limited availability of experimentally verified membrane structures, we focused on the overall soluble character of the generated sequences by predicting TM and soluble residue regions with DeepTMHMM (Hallgren et al., 2022). To realize this comparison, we utilized all 770 sequences from the MeMDLM model test set, yielding a realistic evaluation of soluble residue density.

Figure 4 compares the soluble residue density 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

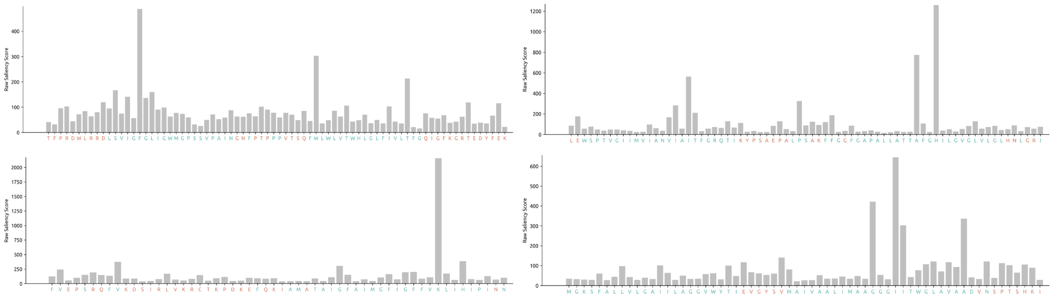

Figure 3: Saliency maps generated for test set sequences.

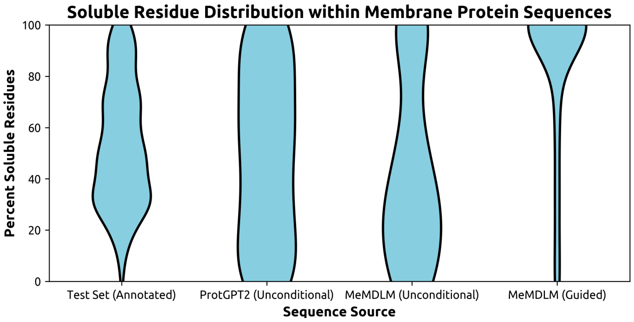

Figure 4: DeepTMHMM-predicted soluble residue density in membrane protein sequences.

that MeMDLM has successfully learned their underlying distribution (Supplementary Table 4). In contrast, ProtGPT2 generates a more uniform distribution of soluble residues, suggesting an overgeneralization of key membrane protein characteristics. Furthermore, unconditionally-generated MeMDLM sequences that are optimized with discrete classifier guidance exhibit an even higher soluble residue density, highlighting the effectiveness of our guidance method. Visualizing randomly selected MeMDLM-generated sequences with AlphaFold3 (Supplementary Figure 6) also confirms the presence of hallmark membrane protein structures, including alpha-helical bundles and central TM regions (Zhang et al., 2015).

As a natural extension of *de novo* design, we scaffolded around TM and soluble motifs of experimentally annotated membrane proteins (Figure 5, Supplementary F.2). We take the entire test set—comprising 770 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. We further apply classifier-guided discrete diffusion to optimize insoluble regions of the test set proteins, observing an increase in soluble residue density.

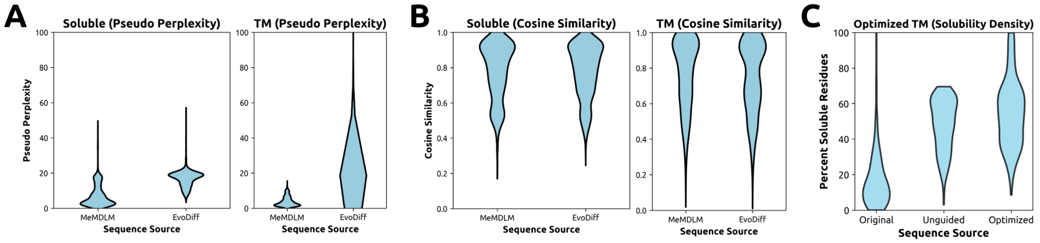

Figure 5: Distribution comparison of reconstruction quality. **A** Pseudo perplexity of soluble and TM regions scaffolded by MeMDLM and EvoDiff. **B** Cosine similarity between embeddings of true and reconstructed sequences from MeMDLM and EvoDiff. **C** DeepTMHMM-predicted soluble residue density in original, unguided, and guided MeMDLM sequences.

5 Discussion

We introduce MeMDLM, the first classifier-guided masked diffusion language model for *de novo* membrane protein generation. By overcoming the sequential bias of autoregressive models, MeMDLM captures long-range dependencies crucial for structural and functional integrity. With LaMBO-2 integration, we enable property-guided optimization, generating soluble and biologically relevant sequences. MeMDLM outperforms existing methods in scaffolding functional motifs and producing structurally realistic membrane proteins. Moving forward, we will experimentally characterize these proteins, assessing TM domain integrity, solubility, and stability to validate MeMDLM's potential for therapeutic development.

REFERENCES

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

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`.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

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.

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*, pp. 2022–04, 2022.

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

Masoud Jelokhani-Niaraki. Membrane proteins: structure, function and motion, 2022.

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.

Andrei L Lomize, Mikhail A Lomize, Shean R Krolicki, and Irina D Pogozheva. Membranome: a database for proteome-wide analysis of single-pass membrane proteins. *Nucleic acids research*, 45 (D1):D250–D255, 2017.

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.

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.

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.

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.

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`.

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.

A Data Curation

A.1 MeMDLM Training Data

Pre-training We queried the UniRef50 database for a random set of 100,000 unique protein sequences containing only the 20 natural amino acids; we only considered sequences shorter than 1,024 residues due to GPU memory limits, and shorter sequences were padded to this maximal length. Sequences were split using the MMSeqs2 easy clustering module with a minimum sequence identity of 30% and a coverage threshold of 50%. The resulting clusters were split to a 80-10-10 ratio into the training set (80,231 sequences, 80.23%), validation set (9,904 sequences, 9.90%), and the testing set (9,865 sequences, 9.87%).

Fine-tuning 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 3.05. 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. The remaining 9,325 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 (7,632 sequences, 81.81%), the validation set (927 sequences, 9.94%), and the testing set (770 sequences, 8.25%).

Value Function We leveraged the same set of 9,329 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.

A.2 Physicochemical Property Prediction Model Training Data

Solubility Prediction We leveraged the same set of 9,329 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 (Thumuluri et al., 2022) 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.

B Physicochemical Property Prediction

B.1 Model Design

Masked Language Model (MLM) ESM-MLM is a fine-tuned encoder that produces membrane-aware protein sequence embedding used as a baseline comparison for the MDLM training task.

We trained a MLM head on top of ESM-2-150M 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 MDLM training. Corrupted sequences were passed into ESM-2-150M to retrieve their output embeddings. The MLM loss function is defined as:

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \mathcal{M}} \log P(x_i | x_{\setminus \mathcal{M}}) \tag{7}$$

where $\mathcal{M}$ represents the set of masked positions in the input sequence, x_i is the true amino acid token at position i, and $x_{\setminus \mathcal{M}}$ denotes the sequence with the masked tokens excluded.

During training, we unfroze the key, query, and value weights in the attention heads of the final three encoder layers. With this training recipe, we augment the pre-existing ESM-2-150M latent space with physicochemical properties of membrane proteins without overfitting on the new sequences. ESM-MLM was tasked to minimize the NLL (equation 7) on 4xA6000 NVIDIA GPUs during training using a learning rate of 5e-3, the Adam optimizer, and a batch size of 8 over 10 epochs.

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). The BCE loss is formally defined as: BCE$(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$

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 (above) 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).

B.2 Representation Quality

We leveraged the trained solubility prediction and membrane localization classifiers to determine if MDLM training retains and augments the original BERT model's representation quality (Table 1).

Model	Solubility (↑)	Membrane Localization (↑)
ESM-2-150M	0.966	0.576
ESM-MLM	0.897	0.584
MeMDLM	0.949	0.541

Table 1: Performance comparison (AUROC) of embeddings in predicting physicochemical properties of MeMDLM test set sequences.

MeMDLM latent embeddings achieve predictive performance that closely parallels SOTA pLM embeddings, which are specifically designed to deliver dense sequence representations. These results demonstrate that MeMDLM accurately captures the biological features underpinning functional membrane proteins.

C Model Design and Implementation

C.1 Masked Diffusion Language Models

We utilized the full MDLM implementation (`https://github.com/kuleshov-group/mdlm`) to design MeMDLM, replacing the DiT backbone with the ESM-2-150M pLM. We leveraged full-parameter training of the ESM backbone during both pre-training and fine-tuning stages. MeMDLM was trained on 4xA6000 NVIDIA GPUs using a learning rate of 3e-4 with cosine warmup, the AdamW optimizer (weight decay of 0.075; betas of 0.9 and 0.999), and a batch size of 8 over five epochs during pre-training and 60 epochs during fine-tuning.

C.2 Value Function

v_θ is the value function used to enable classifier-guided sampling in MeMDLM. To design our value function, we utilize four Transformer encoder layers with two self-attention heads each, applying LayerNorm and dropout ($p = 0.5$) on encoder outputs. A two-layer perceptron with sigmoid activation receives these encoder outputs to obtain per-residue solubility probabilities (predictions). The value function was trained on 2xA6000 NVIDIA GPUs and tasked to minimize the standard BCE loss using a learning rate of 3e-4 with cosine warmup, the AdamW optimizer, and a batch size of 16 over five epochs. With this training setup, the value function achieved a test AUROC of 0.92, demonstrating its reliability for guiding gradient-based edit position selection.

Model	**Train Loss** (↓)	**Val Loss** (↓)	**Test Loss** (↓)	**Test AUROC** (↑)
Value Function	0.2234	0.3044	0.3391	0.9253

Table 2: Training and evaluation performance of the value function (per-reside solubility classifier).

To follow the LaMBO-2 implementation of classifier-based edits, we embedded sequences using the unconditional MeMDLM latent space to train v_θ. Before tokenization, training sequences were further corrupted using the same log-linear corruption scheduler α_t used to train MeMDLM where select amino acid tokens are replaced with the `<mask>` token.

D Protein Language Model Training and Evaluation

	Loss (↓)			**Perplexity** (↓)		
	Train	**Val**	**Test**	**Val**	**Validation**	**Test**
ESM-MLM	0.072	0.072	0.072	1.074	1.074	1.074
ProtGPT2	1.585	–	3.392	4.879	–	29.730
MeMDLM	0.695	2.479	2.230	2.002	7.722	9.285

Table 3: Loss and perplexity comparison across protein language models.

E Protein Sequence Generation

E.1 ProtGPT2

Prepared sequences—split to contain 60 amino acids per line with beginning- and end-of-sequence tags—were passed into the run_clm.py script (`https://huggingface.co/nferruz/ProtGPT2`) to fine-tune the pre-trained ProtGPT2 pLM (Ferruz et al., 2022). Fine-tuning was performed over 5 epochs with a learning rate of 5e-4 and batch size of 2, calculating training loss at every step as the negative log-likelihood loss between logits and labels. The fine-tuned model was used to generate 100 *de novo* membrane protein sequences.

E.2 EvoDiff

We utilized the native EvoDiff codebase (`https://github.com/microsoft/evodiff`) and the provided pre-trained checkpoint `OA_DM_38M()` to infill insoluble and soluble domains of membrane protein sequences. We evaluated EvoDiff's reconstruction quality against MeMDLM's to provide a comparison to SOTA inpainting methods.

E.3 MeMDLM

We generated 100 *de novo* protein sequences of random lengths by inputting sequences consisting of only `<MASK>` tokens into the forward pass of MeMDLM. Next, we scaffolded around TM or soluble motifs by masking specific residues; partially masked sequences were passed through the model for generation. We evaluated MeMDLM against EvoDiff's reconstruction quality. We further applied our optimization scheme to design soluble analogs of our unconditionally generated membrane proteins.

F Generation Quality Evaluation

F.1 *De Novo* Designs

We used AlphaFold3 (AF3) to visualize the structures of naturally occurring and *de novo*-generated membrane protein sequences. Since AF3 was trained primarily on PDB structures, it has an inherent bias to produce more confident predictions for sequences that resemble those in the PDB. To mitigate this bias and motivate a more rigorous comparison, we focused on membrane protein sequences that are not present in the PDB, using them as a benchmark to compare our *de novo* designs, which represent novel topologies not yet captured in natural membrane proteins. We randomly selected UniProt IDs of membrane protein sequences from the Membranome database, ensuring that each selected UniProt ID does not have a corresponding PDB ID (Lomize et al., 2017). Figure 6 shows that ProtGPT2 structures do not exhibit the classic membrane protein architecture, where a TM domain spans the bilayer in the center and soluble regions are located at the intra- and extracellular domains of the protein. However, MeMDLM's structures closely match the expected and naturally occurring topology of membrane proteins, with a clearly defined TM domain in the center of the protein surrounded by soluble regions. We annotated soluble and TM domains using DeepTMHMM.

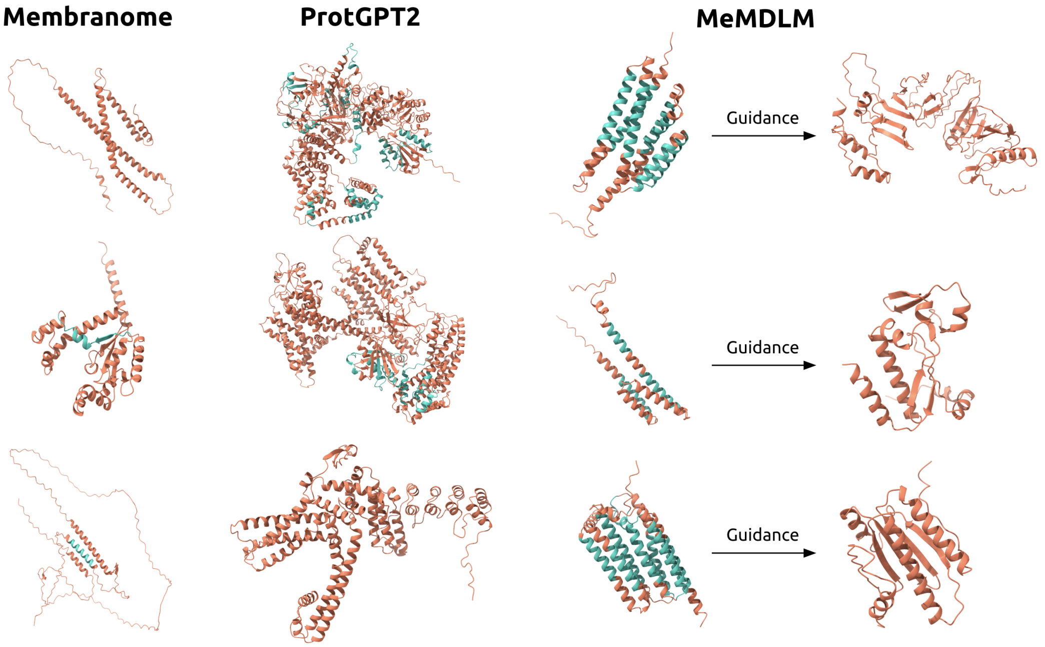

Figure 6: AlphaFold3-predicted structures of natural and *de novo*-generated membrane protein sequences with soluble (orange) and teal (TM) residues annotated.

Metric	**Test Set**	**ProtGPT2**	**MeMDLM**	**MeMDLM**
	(Annotated)	**(Unconditional)**	**(Unconditional)**	**(Guided)**
Soluble Residue Density	53.2 ± 24.1	48.8 ± 33.9	43.2 ± 35.9	86.4 ± 29.0

Table 4: Average soluble residue density in membrane protein sequences.

F.2 Scaffolding Functional Motifs

We compare the cosine similarity between ESM-2-650M embeddings of test set sequences and their reconstructed counterparts, along with the ESM-2-650M pseudo-perplexity of the reconstructed sequence.

	Transmembrane Domain		**Soluble Domain**	
	Pseudo Perplexity	**Cosine Similarity**	**Pseudo Perplexity**	**Cosine Similarity**
MeMDLM	3.819 ± 2.745	0.768 ± 0.193	7.029 ± 6.021	0.778 ± 0.159
EvoDiff	20.554 ± 65.368	0.742 ± 0.200	16.990 ± 4.704	0.777 ± 0.149

Table 5: Reconstruction quality comparison of models scaffolding around TM and soluble motifs of 770 experimental membrane protein sequences that reprsent the MeMDLM model test set.

Table 5 shows that MeMDLM-inpainted sequences not only achieve lower average pseudo-perplexities but also exhibit cosine similarities closely aligned with EvoDiff-based scaffolds across both soluble and TM domains. These results suggest that MeMDLM scaffolds functional motifs with greater confidence while preserving biological relevance.

G Classifier-Guided Sampling Framework

Algorithm 1 Classifier-Guided Sampling with MeMDLM

```
Input: Prior sequence $w_0 = \{w_i\}_{i=1}^{L}$ for $w_i \in \mathcal{V}$, diffusion model $x_\theta$, value function $v_\theta$,
    optimization algorithm $O_\theta$, sampling steps $N$.
Output: Logits $x_s$
for $i \in 1, 2, ..., N$ do:
    $t \sim \text{Uniform}([0, 1])$
    $h, e \leftarrow x_\theta(w_0)$
    $h \leftarrow O_\theta(h, e)$
    $p_\theta(x_0) = \text{SUBS}(h, w_0)$
    Compute transition probability categoricals $q(x_s|x_t)$
    Sample $x_s \sim q(x_s|x_t)$
end for
return $x_s$
```

Algorithm 2 Token-level Optimization via Saliency Maps

```
Inputs: Protein sequence $w$, logits $h$, diffusion model hidden states $e$, diffusion model $x_\theta$, value
    function $v_\theta$, sequence length $l$, optimization steps $S$. Sampling parameters $\lambda, \eta, \epsilon, \tau$.
Output: Optimized logits $h'$.
Initialize Adagrad($\Delta = 0, \eta$) for $\Delta \in \mathbb{R}^{\text{dim}(e)}$
Initialize $\Delta_{\text{saliency}} = 0$ for $\Delta_{\text{saliency}} \in \mathbb{R}^{\text{dim}(e)}$
for $s = 1 \to S$ do
    Initialize SGD ($\Delta_{\text{saliency}}, \eta_{\text{saliency}} = 1$)
    $e' \leftarrow e + \Delta + \Delta_{\text{saliency}}$
    $\mathbf{S}_i = \max\left\{ \left( \sum_{j=1} \left| \nabla_{\Delta_{\text{saliency}}} v_\theta(e')_{ij} \right| \right)^{1/\tau}, \epsilon \right\}$
    Update $\Delta_{\text{saliency}}$ using SGD
    $\mathbf{S}_i = \begin{cases} 0, & \text{if } v_\theta(e')_i \geq 0.5 \\ \mathbf{S}_i, & \text{else} \end{cases}$
    $p_{\text{edit}}(w) = \mathbf{S}_i / \sum \mathbf{S}_i$
    $K_{\text{edit}} = \text{top-k}(p_{\text{edit}}(w), k = \min\{10, p_{\text{edit}}(w) \neq 0\})$
    $\mathbf{m} = \begin{cases} 1, & \text{if } i \in K_{\text{edit}} \\ 0, & \text{else} \end{cases}$
    $e' \leftarrow e + \Delta_{\text{saliency}} \odot \mathbf{m}$
    $h' \leftarrow x_\theta^L(e')$
    $\mathcal{L} = \lambda[D_{\text{KL}}\left[h' \parallel h\right] - \sum_{i=1} v_\theta(e')_i]$
    $h' \leftarrow h' + x_\theta^L(e' + \nabla_\Delta \mathcal{L})$
    Update $\Delta$ using Adagrad with $\nabla_\Delta \mathcal{L}$
end for
$h' \leftarrow x_\theta^L(e' + \Delta)$
return $h'$
```