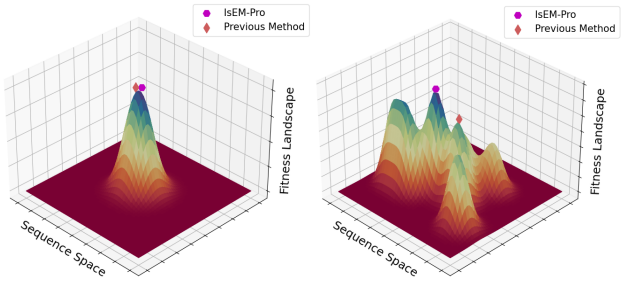
Importance Weighted Expectation-Maximization for Protein Sequence Design

Zhenqiao Song [1] **Lei Li** [1]

Abstract

Designing protein sequences with desired biological function is crucial in biology and chemistry. Recent machine learning methods use a surrogate sequence-function model to replace the expensive wet-lab validation. How can we efficiently generate diverse and novel protein sequences with high fitness? In this paper, we propose IsEM-Pro, an approach to generate protein sequences towards a given fitness criterion. At its core, IsEM-Pro is a latent generative model, augmented by combinatorial structure features from a separately learned Markov random fields (MRFs). We develop an Monte Carlo Expectation-Maximization method (MCEM) to learn the model. During inference, sampling from its latent space enhances diversity while its MRFs features guide the exploration in high fitness regions. Experiments on eight protein sequence design tasks show that our IsEM-Pro outperforms the previous best methods by at least 55% on average fitness score and generates more diverse and novel protein sequences.



(a) Fujiyama landscape. (b) Badlands landscape.

Figure 1. **Protein Fitness Landscape** (distribution of any specific functional property for proteins). Protein may exhibit single-peaked fitness landscape (Fujiyama landscape (a)) or multi-peaked landscape (Badlands landscape (b)) (Kauffman & Weinberger, 1989). In Fujiyama landscape, any method could perform well. However, for the rougher Badlands landscape, previous methods get trapped in a worse local optima while our proposed IsEM-Pro can climb much closer to the global optima through the iterative sampling in the latent space.

1. Introduction

Protein engineering aims to discover protein variants with desired biological function, such as fluorescence intensity (Biswas et al., 2021), enzyme activity (Fox et al., 2007), and therapeutic efficacy (Lagassé et al., 2017). Protein sequences embody their function through spontaneous folding of amino-acid sequences into three dimensional structures (Go, 1983; Chothia, 1984; Starr & Thornton, 2017). The mapping from protein sequence to functional property forms a protein fitness landscape that characterizes the protein functional levels, such as the capability to catalyze reaction or bind a specific ligand (Romero & Arnold, 2009; Ren et al., 2022). Traditional approaches to design a specific protein with desired fitness objective involve obtaining protein variants by random mutagenesis (Labrou, 2010) or recombination in laboratory experiments (Ma et al., 2003). These variants are screened and selected in wet-lab experiments (Arnold, 1998) as illustrated in Figure 2.

However, these approaches requires iterative cycles of random mutagenesis and wet-lab validation, which are both money-consuming and time-intensive. Recent machine learning methods attempt to build a surrogate model of protein fitness landscape to accelerate expensive wet-lab screening (Luo et al., 2021; Meier et al., 2021). How can we efficiently discover satisfactory proteins over the exponentially large discrete space? Ideal protein molecules should be novel, diverse and exhibit high fitness. On one hand, designing novel and diverse protein sequences can uncover new functions and also lead to functional diversification (Singh et al., 2016). On the other hand, in addition to the evolutionary pressure for a protein to conserve specific positions of its sequence for function, the diversification of protein sequences can avoid undesired inter-domain association such as misfolding (Wright et al., 2005).

In this paper, we propose an **I**mportance **s**ampling based **E**xpectation-**M**aximization (EM) method to efficiently de-

[1]Department of Computer Science, University of California, Santa Barbara, California, the United States. Correspondence to: Zhenqiao Song <zhenqiao@ucsb.edu>, Lei Li <leili@cs.ucsb.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

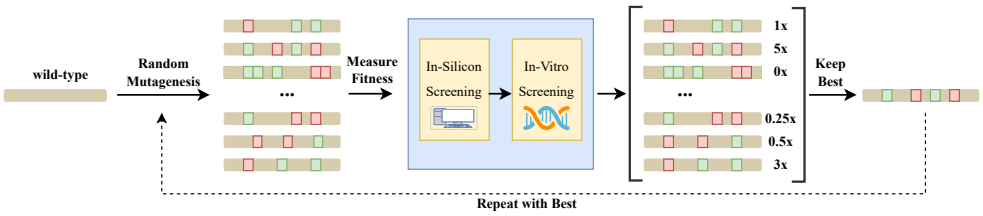


Figure 2. Workflow of traditional protein sequence design. We aim to accelerate this process by directly generating desirable sequences.

sign novel, diverse and desirable **Pro**tein sequences (IsEM-Pro). Specifically, we introduce a latent variable in the generative model to capture the inter-dependencies in protein sequences. Sampling in latent space leads to more diverse candidates and can escape from locally optimal fitness regions. Instead of using standard variational inference models such as variational auto-encoder (VAE) (Kingma & Welling, 2013), we leverage importance sampling inside the EM algorithm to learn the latent generative model. As illustrated in Figure 1, our approach can navigate through multiple local optimum, and yield better overall performance. We further incorporate combinatorial structure of amino acids in protein sequences using Markov random fields (MRFs). It guides the model towards higher fitness landscape, leading to faster uphill path to desired proteins.

We carry out extensive experiments on eight protein sequence design tasks and compare the proposed method with previous strong baselines. The contribution of this paper are listed as follows:

- We propose a structure-enhanced latent generative model for protein sequence design.
- We develop an efficient method to learn the proposed generative model, based on importance sampling inside the EM algorithm.
- Experiments on eight protein datasets with different objectives demonstrate that our IsEM-Pro generates protein sequences with at least 55% higher average fitness score and higher diversity and novelty than previous best methods. Further analyse show that the protein sequences designed by our model can fold stably, giving empirical evidence that our proposed IsEM-Pro has the ability to generate real proteins.

2. Background

In this section, we review protein sequence design and basic variational inference.

2.1. Protein Sequence Design upon Wild-Type

The protein sequence design problem is to search for a sequence with desired property in the sequence space $\mathcal{V}^L$, where $\mathcal{V}$ denotes the vocabulary of amino acids and L denotes the desired sequence length. The target is to find a protein sequence with highest fitness given by a protein fitness function $f : \mathcal{V}^L \to \mathbb{R}$, which can be measured through wet-lab experiments. Wild-type refers to protein occurring in nature. Evolutionary search based methods are widely used (Bloom & Arnold, 2009; Arnold, 2018; Angermueller et al., 2019; Ren et al., 2022). They use wild-type sequence as starting point during iterative search. In this paper, we do not focus on the modification upon the wild-type sequence, but aim to efficiently generate novel and diverse sequences with improved protein functional properties.

2.2. Monte Carlo Expectation-Maximization

A latent generative model assumes data x (e.g. a protein sequence) is generated from a latent variable z. A classic algorithm to learn a latent generative model is Monte Carlo expectation-maximization (MCEM). The optimization procedure for maximizing the log marginal likelihood is to alternate between expectation step (E-step) and maximization step (M-step) (Neal & Hinton, 1998; Jordan et al., 1999). EM directly targets the log marginal likelihood of an observation x by involving a variational distribution $q_\phi(z)$:

$$\begin{aligned}\log p_\theta(x) = E_{q_\phi(z)}[\log p_\theta(x,z) - \log q_\phi(z)] \\ + D_{KL}(q_\phi(z)||p_\theta(z|x))\end{aligned} \tag{1}$$

where $p_\theta(z|x)$ is the true posterior distribution and $p_\theta(x,z) = p_\theta(x|z)p_\theta(z)$ is the joint distribution, composed of the conditional likelihood $p_\theta(x|z)$ and the prior $p_\theta(z)$. In MCEM, E-step samples a set of z from $q_\phi(z)$ to estimate the expectation 1 using Monte Carlo method, and then M-step fits the model parameters θ by maximizing the Monte Carlo estimation (Levine & Casella, 2001). It can be proved that this process will never decease the log marginal likelihood (Dieng & Paisley, 2019). We will develop our method based on this MCEM framework.

3. Proposed Method: IsEM-Pro

In this section, we describe our method in detail. We will first present the probabilistic model and its learning algorithm. To make the learning more efficient, we describe how to uncover and use the potential constraints conveyed in the

protein sequences.

3.1. Problem Formulation

Our goal is to search over a space of discrete protein sequences $\mathcal{V}^L$ – $\mathcal{V}$ consists of 20 amino acids and L is the sequence length – for sequence $x \in \mathcal{V}^L$ that maximizes a given fitness function $f : \mathcal{V}^L \to \mathbb{R}$. Let the fitness value $y = f(x)$, given a predefined threshold λ, we define a conditional likelihood function:

$$P(\mathcal{S}|x) = \begin{cases} 1, & f(x) \geq \lambda, \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathcal{S}$ represents the event that the fitness of x is ideal ($y \geq \lambda$). Using $P_d(x)$ to denote the protein distribution in nature, we assume we have access to a set of observations of x drawn from it. We also assume a class of generative models $P_\theta(x)$ that can be trained with these samples and can approximate P_d well. Since the search space is exponentially large (O(20^L)), random search would be time-intensive. We formulate the protein design problem as generating satisfactory sequences from the posterior distribution $P_\theta(x|\mathcal{S})$:

$$P_\theta(x|\mathcal{S}) = \frac{P_\theta(x)P(\mathcal{S}|x)}{P_\theta(\mathcal{S})} \tag{3}$$

where $P_\theta(\mathcal{S}) = \int_x P_\theta(x)P(\mathcal{S}|x)dx$ is a normalization constant which does not rely on x. Protein sequences generated from $P_\theta(x|\mathcal{S})$ are not only more likely to be real proteins, but also have higher functional scores (i.e., fitness). The higher the λ is, the higher fitness the discovered protein sequences have.

3.2. Probabilistic Model

Directly generating satisfactory sequences from the posterior distribution $P_\theta(x|\mathcal{S})$ is highly efficient compared with randomly search over the exponentially discrete space. However, realizing this idea is difficult as computing $P_\theta(\mathcal{S}) = \int_x P(\mathcal{S}|x)P_\theta(x)dx$ needs an integration over all possible x, which is intractable. Instead, we propose to learn a proposal distribution $Q_\phi(x)$ with learnable parameter ϕ to approximate $P_\theta(x|\mathcal{S})$. In order to find the optimal ϕ of the proposal distribution, we choose to minimize the KL divergence between the posterior distribution $P_\theta(x|\mathcal{S})$ and the proposal distribution $Q_\phi(x)$:

$$\begin{aligned} \phi^* &= \underset{\phi}{\operatorname{argmax}} -D_{KL}(P_\theta(x|\mathcal{S})||Q_\phi(x)) \\ &= \underset{\phi}{\operatorname{argmax}}\, E_{P_\theta(x|\mathcal{S})} \log Q_\phi(x) + \mathcal{H}(\mathcal{P}_\theta) \end{aligned} \tag{4}$$

where $\mathcal{H}(\mathcal{P}_\theta) = -E_{P_\theta(x|\mathcal{S})} \log P_\theta(x|\mathcal{S})$ is the entropy of $P_\theta(x|\mathcal{S})$ and can be dropped because it does not matter ϕ.

Diversity is a key consideration in our protein design procedure, which not only satisfies the diverse nature of species, but also can reduce undesired inter-domain misfolding (Wright et al., 2005). In order to promote the diversity of the designed protein sequences, we introduce a latent variable z into our model to capture the high-order dependencies among amino acids in protein sequences. Thus our final goal is to maximize the log marginal likelihood $\log Q_\phi(x)$ of sequence x from the posterior distribution $P_\theta(x|\mathcal{S})$ integrating over z:

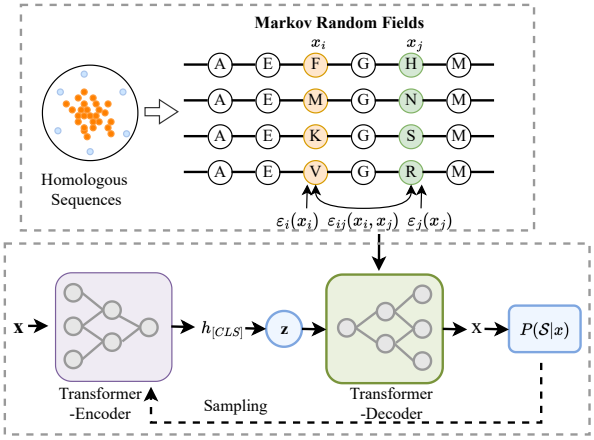


Figure 3. Overall architecture of the proposed IsEM-Pro. The upper half illustrates the Markov random fields which learns the combinatorial structure of amino acids in protein sequences. The bottom half shows the combinatorial structure feature augmented probabilistic model.

$$\begin{aligned} \mathcal{L} &= E_{P_\theta(x|\mathcal{S})} \log Q_\phi(x) \\ &= E_{P_\theta(x|\mathcal{S})}\{E_{R_\omega(z)}[\log Q_\phi(x,z) - \log R_\omega(z)] \\ &\qquad + D_{KL}(R_\omega(z)||Q_\phi(z|x)\} \\ &= E_{P_\theta(x|\mathcal{S})}\mathcal{F}(R_\omega(z), \phi) \end{aligned} \tag{5}$$

where the second equality is the EM objective defined in Equation 1 with approximate posterior distribution $R_\omega(z)$. ω is jointly learned with ϕ by maximizing the above expectation.

For the joint distribution $Q_\phi(x, z) = P(z)Q_\phi(x|z)$, we use standard normal distribution for $P(z)$ and Transformer decoder (Vaswani et al., 2017) for $Q_\phi(x|z)$, which is augmented by the combinatorial structure features introduced in subsection 3.4. The overall architecture is illustrated in Figure 3.

3.3. Importance Weighted EM

To maximize the objective defined in Equation 5, we plan to learn our proposal distribution Q_ϕ through importance sampling based EM, of which the sampling procedure and iterative optimization process can lead to a better estimate (Figure 1 (b)), resulting in novel and diverse proteins with higher fitness.

Since we can not directly sample from $P_\theta(x|\mathcal{S})$ due to the intractable integration factor $P_\theta(\mathcal{S})$, we propose to approximate the expectation using importance sampling with proposal distribution $Q_\phi(x)$ (Neal, 2001; Dieng & Paisley, 2019). Because $\mathcal{S}$ is only conditioned on x, we have $P_\theta(x, z|\mathcal{S}) = \frac{P_\theta(x,z)P(\mathcal{S}|x)}{P_\theta(\mathcal{S})}$. We assume the latent variable z in $P_\theta(x, z)$ and $Q_\phi(x, z)$ are defined on the same latent space $\mathcal{Z}$ and have the same prior $P(z)$. Then the final objective in Equation 5 can be reformulated as:

$$\begin{aligned}\mathcal{L} &= E_{Q_\phi(x)} \frac{P_\theta(x|\mathcal{S})}{Q_\phi(x)} \mathcal{F}(R_\omega(z), \phi) \\ &= E_{Q_\phi(x,z)} \frac{P_\theta(x, z|\mathcal{S})}{Q_\phi(x, z)} \mathcal{F}(R_\omega(z), \phi) \\ &= \frac{1}{\mathcal{C}} E_{Q_\phi(x,z)} \frac{P_\theta(x|z)}{Q_\phi(x|z)} P(\mathcal{S}|x)\mathcal{F}(R_\omega(z), \phi)\end{aligned} \tag{6}$$

The first equality is due to the importance sampling, the second equality holds because $\mathcal{F}(R_\omega(z), \phi)$ is an integration over z and does not rely on z. In the third equality, $\mathcal{C} = P_\theta(\mathcal{S})$ is a constant which does not rely on ϕ and ω, and will be dropped in the following learning process.

We use importance sampling based EM to approximate the above objective (Hastings, 1970; Levine & Casella, 2001) with joint samples $(x_n, z_n) \sim Q_\phi(x, z)$. Specifically, at iteration t, the optimization process can be reformulated as : E-step:

$$\mathcal{L}_t = \frac{\sum_{n=1}^{N} w(x_n, z_n)\mathcal{F}(R_\omega(z_n), \phi)}{\sum_{n=1}^{N} w(x_n, z_n)} \tag{7}$$

M-step:

$$\phi^{(t+1)} = \underset{\phi}{\operatorname{argmax}}\, \mathcal{L}_t \tag{8}$$

where $w(x_n, z_n) = \frac{P_\theta(x_n|z_n)}{Q_{\phi^{(t)}}(x_n|z_n)} P(\mathcal{S}|x_n)$ is the unnormalized importance weight, and N is the sample size.

Through the combined sampling procedure and iterative optimization process, we can obtain a good proposal distribution Q_ϕ, from which we can generate satisfactory sequences with small time cost.

3.4. Guiding Model Climbing through Combinatorial Structure

As shown in previous work, the combinatorial structure of amino acids in protein sequences can be learned from a generative graphical model Markov random fields (MRFs) fitted on the sequences from the same family (Hopf et al., 2017; Luo et al., 2021). These structure constraints are the results of the evolutionary process under natural selection and may reveal clues on which amino-acid combinations are more favorable than others. Thus we incorporate these features into our model to guide it towards higher fitness landscape to faster find desired protein sequences.

Given a protein sequence $x = (x_1, x_2, .., x_L)$ with L amino acids, the generative model generates it with likelihood $P_L(x) = \frac{\exp(E(x))}{Z}$ where $Z = \int_x \exp(E(x))dx$ is a normalization constant and E(x) is the corresponding energy function, which is defined as the sum of all pairwise constraints and single-site constraints as follows:

$$E(x) = \sum_{i=1}^{L} \varepsilon_i(x_i) + \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \varepsilon_{ij}(x_i, x_j) \tag{9}$$

where $\varepsilon_i(x_i)$ denotes the single-site constraint of x_i at position i and $\varepsilon_{ij}(x_i, x_j)$ denotes the pairwise constraint of x_i and x_j at position i, j. The above graphical model is illustrated in the upper half of Figure 3.

We train the model following CCMpred (Seemayer et al., 2014) using a pseudo-likelihood $\hat{P}_L(x)$ (provided in Appendix E) combined with Ridge regularization to make the learning of $P_L(x)$ easier. But different from them, we additionally add a lasso regularizer to the training objective to make the graph sparse, of which the regularization coefficients are set to the same values as the ridge regularizer:

$$\begin{aligned}L &= \sum_x \log \hat{P}_L(x) - \mathcal{L}(\varepsilon) - \mathcal{R}(\varepsilon) \\ \mathcal{L}(\varepsilon) &= \lambda_{\text{single}} \sum_{i=1}^{L} ||\varepsilon_i||_1^1 + \lambda_{\text{pair}} \sum_{i,j=1, i\neq j} ||\varepsilon_{ij}||_1^1 \\ \mathcal{R}(\varepsilon) &= \lambda_{\text{single}} \sum_{i=1}^{L} ||\varepsilon_i||_2^2 + \lambda_{\text{pair}} \sum_{i,j=1, i\neq j} ||\varepsilon_{ij}||_2^2\end{aligned} \tag{10}$$

where $\varepsilon_i = [\varepsilon_i(a_1), \varepsilon_i(a_2), ..., \varepsilon_i(a_{20})]$ is the vector of the single-site constraints of the 20 amino acids at position i, and $\varepsilon_{ij} = [\varepsilon_{ij}(a_1, a_2), \varepsilon_{ij}(a_1, a_3), ..., \varepsilon_{ij}(a_L, a_{L-1})]$ is the vector of all possible pairwise constraints at position i, j. Following (Kamisetty et al., 2013), we set $\lambda_{\text{single}} = 1$ and $\lambda_{\text{pair}} = 0.2 * (L - 1)$.

After training the MRFs, we can encode a protein sequence x with the learned constraints. Specifically, we first encode the i-th amino acid by concatenating its corresponding single-site constraint as well as the possible pairwise ones:

$$\begin{aligned}\boldsymbol{\varepsilon_i}(x_i) &= [\varepsilon_i(x_i), \varepsilon_{i1}(x_i, a_{1.}), ..., \varepsilon_{iL}(x_i, a_{L.})] \\ \varepsilon_{ij}(x_i, a_{j.}) &= [\varepsilon_{ij}(x_i, a_1), \varepsilon_{ij}(x_i, a_2), ..., \varepsilon_{ij}(x_i, a_{20})]\end{aligned} \tag{11}$$

where $\varepsilon_{ij}(x_i, a_{j.})$ gathers the 20 amino acids for any position $j \neq i$. Then we map $\boldsymbol{\varepsilon_i}(x_i)$ to the amino-acid embedding space with trainable parameter W_ε, and add the mapped vector to the original amino-acid embedding $e(x_i)$ to get the final feature vector as our model input:

$$\begin{aligned}&\hat{e}(x_i) = e(x_i) + W_\varepsilon * \boldsymbol{\varepsilon_i}(x_i) \\ &H_0 = \tilde{z}, \quad H_i = \hat{e}(x_{i-1}) \text{ for } 1 \leq i < L\end{aligned} \tag{12}$$

which means for the autoregressive Transformer decoder, the first token input is set the sampled latent vector $\tilde{z}$ and the input for other position is set to the combinatorial structure augmented feature vector of last token.

At iteration t in MCEM, the learning process for the combinatorial structure enhanced latent generative model becomes:
E-step:

$$\mathcal{L}_t = \frac{\sum_{n=1}^{N} w(x_n, z_n)\mathcal{F}(R_\omega(z_n), \phi; \boldsymbol{\varepsilon})}{\sum_{n=1}^{N} w(x_n, z_n)} \tag{13}$$

M-step:

$$\phi^{(t+1)} = \underset{\phi}{\operatorname{argmax}}\, \mathcal{L}_t \tag{14}$$

$\boldsymbol{\varepsilon}$ is fixed during latent generative model learning and we omit it in the following parts to make description simple. The overall learning algorithm is given in Appendix B.2.

4. Experiments

In this section, we conduct extensive experiments to validate the effectiveness of our proposed IsEM-Pro on protein sequence design task.

4.1. Implementation Details

Our model is built based on Transformer (Vaswani et al., 2017) with 6-layer encoder initialized by ESM-2 (Lin et al., 2022)[1] and 2-layer decoder with random initialization, of which the encoder parameters are fixed during training process. Thus the MRFs features are only incorporated in decoder. The model hidden size and feed-forward hidden size are set to 320 and 1280 respectively as ESM-2. We use the [CLS] representation from the last layer of encoder to calculate the mean and variance vectors of the latent variable through single-layer mapping. Then the sampled latent vector is used as the first token input of decoder. The latent vector size is correspondingly set to 320. We first train a VAE model as P_θ for 30 epochs and $\phi^{(0)}$ is initialized by θ. The number of iterative process in the importance sampling based VEM is set to 10. The protein combinatorial structure constraints $\boldsymbol{\varepsilon}$ are learned on the training sequences for each dataset instead of real multiple sequence alignments (MSAs) to keep a fair comparison.

The mini-batch size and learning rate are set to $4,096$ tokens and 1e-5 respectively. The model is trained with 1 NVIDIA RTX A6000 GPU card. We apply Adam algorithm (Kingma & Ba, 2014) as the optimizer with a linear warm-up over the first $4,000$ steps and linear decay for later steps. We randomly split each dataset into training/validation sets with the ratio of 9:1. We run all the experiments for five times and report the average scores. More experimental settings are given in Appendix B.1.

In inference, we design protein sequences by taking the wild-type as encoder input and the latent vector is sampled from prior distribution $P(z)$. The sequences are decoded using sampling strategy with top-5. The candidate number is set to K=128 following the setting of Jain et al. (2022) on the GFP dataset. The code is available at `https://github.com/JocelynSong/IsEM-Pro.git`.

4.2. Datasets

Following Ren et al. (2022), we evaluate our method on the following eight protein engineering benchmarks:
(1) **Green Fluorescent Protein (avGFP)**: The goal is to design sequences with higher log-fluorescence intensity values. We collect data following Sarkisyan et al. (2016). (2) **Adeno-Associated Viruses (AAV)**: The target is to generate amino-acid segment (position $561-588$) for higher gene therapeutic efficiency. We collect data following Bryant et al. (2021). (3) **TEM-1 β-Lactamase (TEM)**: The goal is to design high thermodynamic-stable sequences. We merge the data from Firnberg et al. (2014). (4) **Ubiquitination Factor Ube4b (E4B)**: The objective is to design sequences with higher enzyme activity. We gather data following Starita et al. (2013). (5) **Aliphatic Amide Hydrolase (AMIE)**: The goal is to produce amidase sequences with higher enzyme activity. We merge data following Wrenbeck et al. (2017). (6) **Levoglucosan Kinase (LGK)**: The target is to optimize LGK protein sequences with improved enzyme activity. We collect data following Klesmith et al. (2015). (7) **Poly(A)-binding Protein (Pab1)**: The goal is to design sequences with higher binding fitness to multiple adenosine monophosphates. We gather data following Melamed et al. (2013). (8) **SUMO E2 Conjugase (UBE2I)**: We aim to find human SUMO E2 conjugase with higher growth rescue rate. Data are obtained following Weile et al. (2017). The detailed data statistics, including protein sequence length, data size and data source are provided in Appendix A.

4.3. Baseline Models

We compare our method against the following representative baselines: (1) **CMA-ES** (Hansen & Ostermeier, 2001) is a famous evolutionary search algorithm. (2) **FBGAN** proposed by Gupta & Zou (2019) is a novel feedback-loop architecture with generative model GAN. (3) **DbAS** (Brookes & Listgarten, 2018) is a probabilistic modeling framework and uses adaptive sampling algorithm. (4) **CbAS** (Brookes et al., 2019) improves on DbAS by conditioning on the desired properties. (5) **PEX** proposed by Ren et al. (2022) is a model-guided sequence design algorithm using proximal exploration. (6) **GFlowNet-AL** (Jain et al., 2022) applies GFlowNet to design biological sequences. We use the im-

[1] https://dl.fbaipublicfiles.com/fairesm/models/esm2_t6_8M_UR50D.pt

Models	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
CMA-ES	4.492	−3.417	0.375	−0.768	−8.224	−0.077	0.164	2.461	−0.624
FBGAN	1.251	−4.227	0.006	0.369	−2.410	−1.206	0.029	0.208	−0.747
DbAS	3.548	4.327	0.003	−1.286	−2.658	−1.148	1.524	3.088	0.924
CbAS	3.550	4.336	0.106	−1.000	−1.306	−0.362	1.842	3.263	1.303
PEX	3.764	3.265	0.121	5.019	−0.474	0.007	1.153	1.995	1.856
GFlowNet-AL	5.062	1.205	1.552	3.155	0.059	0.027	2.168	3.576	2.101
ESM-Search	2.610	−5.099	0.148	−1.860	−2.351	−0.029	1.406	3.244	−0.241
IsEM-Pro	**6.185**	**4.813**	**1.850**	**5.737**	**0.062**	**0.035**	**2.923**	**4.536**	**3.267**
– w/o ESM	1.214	−4.313	0.005	−1.352	−6.376	−0.225	0.072	1.843	−1.141
– w/o ISEM	4.708	1.130	0.708	0.046	−2.335	−0.077	1.913	0.475	1.342
– w/o MRFs	4.376	1.008	0.952	0.045	−1.771	−0.012	1.652	2.418	1.083
– w/o LV	4.274	2.251	0.078	−1.612	−2.266	−0.931	0.041	−0.262	0.196

Table 1. Maximum fitness scores (MFS) of all methods on eight datasets. Higher values indicate better functional properties in the dataset. Our proposed IsEM-Pro achieves the highest fitness scores on all datasets.

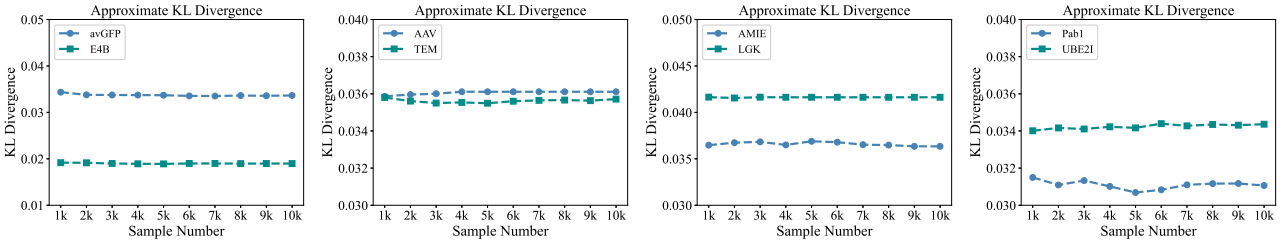
Models	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
CMA-ES	**225.12**	23.50	261.60	86.81	283.90	317.08	61.16	140.92	175.01
FBGAN	0.64	8.31	0.46	3.87	33.87	17.35	3.07	3.00	8.82
DbAS	3.04	3.00	3.67	5.94	1.32	2.30	4.05	11.80	4.33
CbAS	1.31	3.01	7.03	7.09	6.01	6.15	9.86	22.73	8.23
PEX	6.83	4.35	10.26	5.22	7.56	13.24	5.33	10.32	7.88
GFlowNet-AL	224.78	**25.57**	**266.43**	43.62	219.84	212.25	37.13	49.79	134.92
ESM-Search	3.79	3.58	11.56	3.82	3.83	3.78	5.71	6.59	5.33
IsEM-Pro	218.62	22.92	202.09	**91.35**	**293.30**	**405.99**	**68.27**	122.66	**178.15**
– w/o ESM	204.21	13.87	194.78	7.90	276.88	362.98	3.30	119.87	147.96
– w/o ISEM	122.15	22.91	70.12	86.17	145.74	169.17	60.22	13.05	153.09
– w/o MRFs	217.35	17.02	225.88	84.64	268.07	381.66	66.26	**143.29**	175.52
– w/o LV	22.70	5.26	10.12	5.24	8.65	20.28	0.67	2.98	9.48

Table 2. Diversity scores of all models on eight datasets. Higher values indicate more diverse protein sequences. Our IsEM-Pro achieves the highest average diversity scores over the eight protein datasets.

Models	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
CMA-ES	221.55	22.73	269.25	93.78	256.07	415.63	59.35	128.10	183.30
FBGAN	0.05	2.76	0.08	0.63	57.87	39.36	0.75	0.80	12.43
DbAS	1.01	3.01	1.47	1.09	1.12	1.63	1.64	2.05	1.66
CbAS	4.02	3.03	2.06	1.90	1.33	1.09	2.71	2.95	1.92
PEX	3.59	1.88	8.57	4.08	4.50	10.53	3.63	10.24	5.87
GFlowNet-AL	221.95	22.83	266.99	86.78	316.79	412.58	61.33	143.28	191.56
ESM-Search	1.50	1.83	7.32	1.21	0.92	0.91	5.46	6.14	3.16
IsEM-Pro	**226.31**	**23.81**	**270.27**	**96.57**	**332.23**	**420.93**	**70.09**	**153.27**	**199.18**
– w/o ESM	198.08	9.25	176.20	3.79	264.36	340.62	1.49	110.53	138.04
– w/o ISEM	195.85	16.36	244.05	85.81	306.22	382.12	53.01	99.51	180.40
– w/o MRFs	207.59	9.53	221.49	90.81	244.14	327.14	58.44	129.75	161.11
– w/o LV	16.67	0.89	4.39	0.48	3.98	9.64	0.21	1.76	4.75

Table 3. Novelty scores of all models on eight datasets. Higher values indicate more novel protein sequences. Our IsEM-Pro achieves the highest novelty scores on all the datasets.

plementations of CMA-ES, DbAS and CbAS provided in Trabucco et al. (2022) and for other baselines, we apply their released codes. To better analyze the influence of different components in our model, we also conduct ablation tests as follows: (1) **IsEM-Pro-w/o-ESM** removes ESM-2 as encoder initialization. (2) **IsEM-Pro-w/o-ISEM** removes iterative optimization process. (3) **IsEM-Pro-w/o-MRFs** removes MRFs features and iterative optimization process. (4) **IsEM-Pro-w/o-LV** removes latent variable, MRFs features and iterative optimization process. (5) **ESM-Search** samples sequences from the softmax distribution obtained by finetuning ESM-2 on the protein datasets and taking the



(a) avGFP and E4B (a) AAV and TEM (b) AMIE and LGK (c) Pab1 and UBE2I

Figure 4. Approximate KL divergence on eight protein datasets. It shows the variance of KL divergence is very small over different sample size for all datasets, giving empirical evidence that when we sample from the ultimate $Q_\phi(x)$, it has minor difference compared with sampling from the posterior distribution $P_\theta(x|\mathcal{S})$.

wild-type as input.

4.4. Evaluation Metrics

We use three automatic metrics to evaluate the performance of the designed sequences: (1) **MFS**: Maximum fitness score. The oracle model adopted to evaluate MFS is described in Appendix B.1; (2) **Diversity** proposed by Jain et al. (2022) is used to evaluate how different the designed candidates are from each other; (3) **Novelty** proposed by Jain et al. (2022) is used to evaluate how different the proposed candidates are from the sequences in training data.

4.5. Main Results

Table 1, 2 and 3 respectively report the maximum fitness scores, diversity scores and novelty scores of all models.

IsEM-Pro achieves the highest fitness scores on all protein families and outperforms the previous best method GFlowNet-AL by 55% on average (Table 1). The reasons are two-folds. On one hand, the importance sampling based VEM can help our model to navigate to a better region instead of getting trapped in a worse local optima. On the other hand, the combinatorial structure features help to recognize the preferred mutation patterns which have higher success rate under the nature selection pressure, potentially leading to sequences with higher fitness scores.

IsEM-Pro achieves the highest average diversity score over the eight tasks (Table 2). Our model gains the highest diversity on 4 out of 8 tasks while GFlowNet-AL gains 2o and CMA-ES gains 1. It indicates that though involving combinatorial structure constraints can give the guidance for preferred protein patterns, it might also limit the sequence design to these patterns to some extent. The involved latent variable can capture complex inter-dependencies among amino acids, which benefits for more diverse protein design.

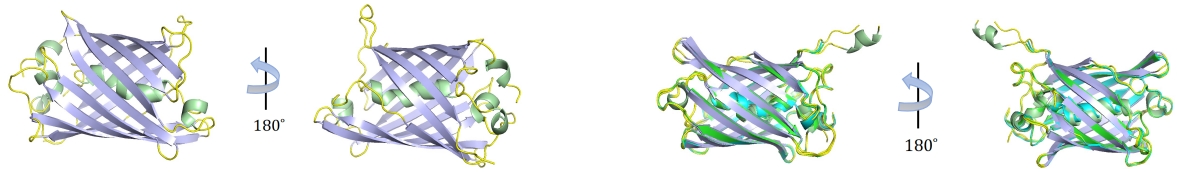
IsEM-Pro can design more novel protein sequences on all datasets (Table 3). Our model achieves higher novelty scores on all datasets due to the reason that more new samples are involved during the importance sampling based iterative optimization process, which is beneficial for more novel protein design.

4.6. Ablation Study

Bottom halves of Table 1, 2 and 3 report the results of ablation tests. IsEM-Pro-w/o-MRFs improves the average diversity and novelty scores by as much as 33x compared with IsEM-Pro-w/o-LV, which demonstrates that introducing a latent variable can significantly help to generate diverse proteins. IsEM-Pro-w/o-MRFs achieves higher maximum fitness scores than IsEM-Pro-w/o-ESM on all datasets, validating that adopting a pretrained protein language model as the encoder helps to design more satisfactory protein sequences. However, directly finetuning ESM-2 to sample candidates (ESM-Search) drops 1.3 points on average fitness score compared with taking ESM-2 as an encoder (IsEM-Pro-w/o-MRFs), demonstrating that ESM-2 is not suitable for direct sequence design. Incorporating combinatorial structure features can further improve the fitness of the designed proteins (IsEM-Pro-w/o-ISEM V.S. IsEM-Pro-w/o-MRFs), based on which learning the proposal distribution by importance sampling based VEM can better promote more desirable, diverse and novel protein generation. We also provide the model performance with a fully-connected decoder (non-autoregressive decoder) instead of the current autoregressive one in Appendix F.2. It shows the non-autoregressive decoder can not generate good candidates for protein with longer sequences such as LGK (439 amino acids).

5. Analysis

In this section, we will make rigorous analyse to demonstrate the effectiveness of our method from different aspects.



(a) 3-D visualization of sequence with highest fitness. (b) Superposition of 5 fluorescent proteins in top-5 templates.

Figure 5. 3-D visualization of a designed sequence of green fluorescent protein, validating that our model is highly likely to design a real fluorescent protein.

Methods	MFS	Diversity	Novelty
Latent-Add	4.102	218.26	209.85
Latent-Memory	4.040	**219.00**	211.38
IsEM-Pro	**6.185**	218.62	**226.31**

Table 4. Results of different schemes of introducing a latent variable with a pretrained encoder evaluated on avGFP dataset. It shows that our method, which takes the latent representation as the first token input of decoder, achieves a higher fitness and novelty scores though with a mild decrease on diversity.

5.1. Approximate KL Divergence

To validate how close the proposal distribution $Q_\phi(x)$ is to the posterior distribution $P_\theta(x|\mathcal{S})$, we calculate the KL divergence through Monte Carlo approximation. Here we calculate the KL divergence between $Q_\phi(x)$ and $P_\theta(x|\mathcal{S})$ as we have proven in Lemma $C.1$ (provided in Appendix C.2) that the sampling difference between these two distributions can be bounded under this divergence. We leverage an unbiased and low-variance estimator (proof shown in Appendix C.1) to approximate KL divergence as follows:

$$D_{KL}(Q_\phi(x)||P_\theta(x|\mathcal{S})) = E_{Q_\phi(x)}[r(x) - 1 - \log r(x)] \quad (15)$$

where $r(x) = \frac{P_\theta(x|\mathcal{S})}{Q_\phi(x)}$. The approximate KL divergence on eight protein datasets over 1k-10k samples are illustrated in Figure 4. From the figure, we can see that the variance of KL divergence is very small over different sample size for all datasets. Besides, the KL divergence finally arrives at a small value, such as 0.018 for E4B and 0.033 for avGFP. It gives empirical evidence that when we sample from the ultimate $Q_\phi(x)$, it has minor difference compared with sampling from the posterior distribution $P_\theta(x|\mathcal{S})$.

5.2. Effect of VAE Implementation Method

Next, we study the effect of different implementation schemes of involving a latent variable with a pretrained encoder. Some works have tried adding the latent representation to the original embedding layer (Latent-Add) or using it as an additional memory (Latent-Memory) when adopting a pretrained language model as encoder (Li et al., 2020). We also implement our model with these two schemes, and evaluate the model performance on avGFP dataset. Table 4 shows that our method, which takes the latent representation as the first token input of decoder, achieves a higher fitness and novelty scores though with a mild decrease on diversity. We also provide the conditional VAE (CVAE) performance in Appendix F.1, which shows decreased performance. It demonstrates that iterative sampling inside the EM algorithm with explicit fitness constraints can lead to better proteins.

5.3. Case Study

To gain an insight on how well the designed proteins are, we analyze the generated avGFP sequence with highest fitness in detail using Phyre2 tool (Kelley et al., 2015). Figure 5 (a) illustrates the generated variant can fold stably. According to the software, the most similar protein is Cytochrome b562 integral fusion with enhanced green fluorescent protein (EGFP) (Edwards et al., 2008). There are 227 residues (96% of the candidate sequence) have been modeled with 100.0% confidence by using this protein as template. Details are given in Appendix D. Figure 5(b) visualizes the superposition of the top-5 most similar templates to our sequence in the protein data bank, which are all fluorescent proteins and show highly consistent structure in most regions, validating that our model can design a real fluorescent protein.

6. Related Work

Machine Learning for Protein Fitness Landscape Prediction. Machine learning has been increasingly used for modeling protein fitness landscape, which is crucial for protein engineering. Some work leverage co-evolution information from multiple sequence alignments to predict fitness scores (Kamisetty et al., 2013; Luo et al., 2021). Melamed et al. (2013) propose to construct a deep latent generative model to capture higher-order mutations. Meier et al. (2021) propose to use pretrained protein language models to enable zero-shot prediction. The learned protein landscape models can be used to replace the expensive wet-lab validation to screen enormous designed sequences (Rao et al., 2019; Ren et al., 2022).

Methods for Protein Sequence Design. Protein sequence

design has been studied with a wide variety of methods, including traditional directed evolution (Arnold, 1998; Dalby, 2011; Packer & Liu, 2015; Arnold, 2018) and machine learning methods. The mainly used machine learning algorithms include reinforcement learning (Angermueller et al., 2019; Jain et al., 2022), Bayesian optimization (Belanger et al., 2019; Moss et al., 2020; Terayama et al., 2021), search using deep generative models (Brookes & Listgarten, 2018; Brookes et al., 2019; Madani et al., 2020; Kumar & Levine, 2020; Das et al., 2021; Hoffman et al., 2022; Melnyk et al., 2021; Ren et al., 2022) adaptive evolution methods (Hansen, 2006; Swersky et al., 2020; Sinai et al., 2020) as well as likelihood-free inference (Zhang et al., 2021).

Different from the existing work, we propose importance sampling based VEM to learn a latent generative model which is enhanced by combinatorial structure features of protein space. The whole framework can not only help the generative model to climb to a better region in either Fujiyama landscape or Badlands landscape (Kauffman & Weinberger, 1989), but also significantly promote design diversity and novelty.

7. Discussion

In this section, we discuss some architecture choice and limitation of this paper.

We opt for MCEM over the standard VAE to learn the latent generative model due to our belief that the standard VAE has certain limitations. The major limitation of the standard VAE is that it maximizes the likelihood of observed data, say protein sequence, but higher likelihood does not necessarily associate with higher fitness of a protein. Our IsEM-Pro tackles this by explicitly modelling fitness constraints and amino acid correlation in a protein sequence. As shown in Table 1, 2 and 3, the fitness scores of standard VAE (IsEM-Pro-w/o-MRFs) decrease a lot compared with our IsEM-Pro and the diversity and novelty scores also slightly decrease. Besides, as Dieng & Paisley (2019) analyzed, VAE amortizes the cost of inference by using a recognition network to parameterize the variational family, which introduces an amortization gap and leads to approximate posteriors of reduced expressivity due to the problem known as posterior collapse. Instead, EM directly maximizes the log marginal likelihood of the data and each iteration in EM is guaranteed to increase the log marginal likelihood from the previous iteration (Dieng & Paisley, 2019). Therefore, using EM in the context of deep generative models could lead to better performance. Additionally, standard VAE attempts to perform variational inference through a direct, discriminative mapping from data observations to approximate posterior parameters. Though generative models can adapt to accommodate sub-optimal approximate posteriors, it's likely limited to direct inference mapping, leading to being trapped in a worse local optima (Marino et al., 2018). Instead, EM guarantees that the log marginal likelihood of the data keeps increasing in each iteration, helping our model climb much closer to the global optima.

One limitation of this work is, although our model has demonstrated promising results, the designed protein sequences have not undergone wet-lab testing and there might be some uncertainty correlated with the oracle model. The results reported in our paper are averaged over five runs, which can accommodate some variance on this metric. Furthermore, all fitness data used in our paper are obtained from wet-lab experiments, ensuring that the fitness values of the training datasets are realistic. To reduce the cost of wet-lab validation and select good protein candidates, we evaluate the designed sequences using the oracle model trained on real protein sequences following (Ren et al., 2022; Jain et al., 2022; Rao et al., 2019), which may associate with some uncertainties. While there is currently no perfect automatic metric available, we believe that new methods should be encouraged in the field of computational biology. With time, we are confident that the field will continue to develop and mature.

8. Conclusion

This paper proposes IsEM-Pro, a latent generative model for protein sequence design, which incorporates additional combinatorial structure features learned by MRFs. We use importance weighted EM to learn the model, which can not only enhance design diversity and novelty, but also lead to protein sequences with higher fitness. Experimental results on eight protein sequence design tasks show that our method outperforms several strong baselines on all metrics.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. We are also grateful to Siqi Ouyang, Yujian Liu, Jingjing Xu, Danqing Wang and Antonis Antoniades for their great suggestions. This work is partially supported by UCSB Faculty Research Award.

References

Angermueller, C., Dohan, D., Belanger, D., Deshpande, R., Murphy, K., and Colwell, L. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*, 2019.

Arnold, F. H. Design by directed evolution. *Accounts of chemical research*, 31(3):125–131, 1998.

Arnold, F. H. Directed evolution: bringing new chemistry to

life. *Angewandte Chemie International Edition*, 57(16): 4143–4148, 2018.

Belanger, D., Vora, S., Mariet, Z., Deshpande, R., Dohan, D., Angermueller, C., Murphy, K., Chapelle, O., and Colwell, L. Biological sequences design using batched bayesian optimization. 2019.

Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., and Church, G. M. Low-n protein engineering with data-efficient deep learning. *Nature methods*, 18(4):389–396, 2021.

Bloom, J. D. and Arnold, F. H. In the light of directed evolution: pathways of adaptive protein evolution. *Proceedings of the National Academy of Sciences*, 106(supplement_1): 9995–10000, 2009.

Brookes, D., Park, H., and Listgarten, J. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, pp. 773–782. PMLR, 2019.

Brookes, D. H. and Listgarten, J. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*, 2018.

Bryant, D. H., Bashir, A., Sinai, S., Jain, N. K., Ogden, P. J., Riley, P. F., Church, G. M., Colwell, L. J., and Kelsic, E. D. Deep diversification of an aav capsid protein by machine learning. *Nature Biotechnology*, 39(6):691–696, 2021.

Chothia, C. Principles that determine the structure of proteins. *Annual review of biochemistry*, 53(1):537–572, 1984.

Csiszár, I. and Körner, J. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

Dalby, P. A. Strategy and success for the directed evolution of enzymes. *Current opinion in structural biology*, 21(4): 473–480, 2011.

Das, P., Sercu, T., Wadhawan, K., Padhi, I., Gehrmann, S., Cipcigan, F., Chenthamarakshan, V., Strobelt, H., Dos Santos, C., Chen, P.-Y., et al. Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nature Biomedical Engineering*, 5(6):613–623, 2021.

Dieng, A. B. and Paisley, J. Reweighted expectation maximization. *arXiv preprint arXiv:1906.05850*, 2019.

Edwards, W. R., Busse, K., Allemann, R. K., and Jones, D. D. Linking the functions of unrelated proteins using a novel directed evolution domain insertion method. *Nucleic acids research*, 36(13):e78–e78, 2008.

Firnberg, E., Labonte, J. W., Gray, J. J., and Ostermeier, M. A comprehensive, high-resolution map of a gene's fitness landscape. *Molecular biology and evolution*, 31 (6):1581–1592, 2014.

Fox, R. J., Davis, S. C., Mundorff, E. C., Newman, L. M., Gavrilovic, V., Ma, S. K., Chung, L. M., Ching, C., Tam, S., Muley, S., et al. Improving catalytic function by prosar-driven enzyme evolution. *Nature biotechnology*, 25(3):338–344, 2007.

Go, N. Theoretical studies of protein folding. *Annual review of biophysics and bioengineering*, 12(1):183–210, 1983.

Gupta, A. and Zou, J. Feedback gan for dna optimizes protein functions. *Nature Machine Intelligence*, 1(2): 105–111, 2019.

Hansen, N. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation*, pp. 75–102, 2006.

Hansen, N. and Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. 1970.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Hoffman, S. C., Chenthamarakshan, V., Wadhawan, K., Chen, P.-Y., and Das, P. Optimizing molecules using efficient queries from property evaluations. *Nature Machine Intelligence*, 4(1):21–31, 2022.

Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P., Springer, M., Sander, C., and Marks, D. S. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

Jain, M., Bengio, E., Hernandez-Garcia, A., Rector-Brooks, J., Dossou, B. F., Ekbote, C. A., Fu, J., Zhang, T., Kilgour, M., Zhang, D., et al. Biological sequence design with gflownets. In *International Conference on Machine Learning*, pp. 9786–9801. PMLR, 2022.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kamisetty, H., Ovchinnikov, S., and Baker, D. Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39): 15674–15679, 2013.

Kauffman, S. A. and Weinberger, E. D. The nk model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology*, 141(2):211–245, 1989.

Kelley, L. A., Mezulis, S., Yates, C. M., Wass, M. N., and Sternberg, M. J. The phyre2 web portal for protein modeling, prediction and analysis. *Nature protocols*, 10(6): 845–858, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Klesmith, J. R., Bacik, J.-P., Michalczyk, R., and Whitehead, T. A. Comprehensive sequence-flux mapping of a levoglucosan utilization pathway in e. coli. *ACS synthetic biology*, 4(11):1235–1243, 2015.

Kumar, A. and Levine, S. Model inversion networks for model-based optimization. *Advances in Neural Information Processing Systems*, 33:5126–5137, 2020.

Labrou, N. E. Random mutagenesis methods for in vitro directed enzyme evolution. *Current Protein and Peptide Science*, 11(1):91–100, 2010.

Lagassé, H. D., Alexaki, A., Simhadri, V. L., Katagiri, N. H., Jankowski, W., Sauna, Z. E., and Kimchi-Sarfaty, C. Recent advances in (therapeutic protein) drug development. *F1000Research*, 6, 2017.

Levine, R. A. and Casella, G. Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

Li, C., Gao, X., Li, Y., Peng, B., Li, X., Zhang, Y., and Gao, J. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*, 2020.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

Luo, Y., Jiang, G., Yu, T., Liu, Y., Vo, L., Ding, H., Su, Y., Qian, W. W., Zhao, H., and Peng, J. Ecnet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature communications*, 12(1): 1–14, 2021.

Ma, J. K., Drake, P. M., and Christou, P. The production of recombinant pharmaceutical proteins in plants. *Nature reviews genetics*, 4(10):794–805, 2003.

Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., and Socher, R. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Marino, J., Yue, Y., and Mandt, S. Iterative amortized inference. In *International Conference on Machine Learning*, pp. 3403–3412. PMLR, 2018.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

Melamed, D., Young, D. L., Gamble, C. E., Miller, C. R., and Fields, S. Deep mutational scanning of an rrm domain of the saccharomyces cerevisiae poly (a)-binding protein. *Rna*, 19(11):1537–1551, 2013.

Melnyk, I., Das, P., Chenthamarakshan, V., and Lozano, A. Benchmarking deep generative models for diverse antibody sequence design. *arXiv preprint arXiv:2111.06801*, 2021.

Moss, H., Leslie, D., Beck, D., Gonzalez, J., and Rayson, P. Boss: Bayesian optimization over string spaces. *Advances in neural information processing systems*, 33: 15476–15486, 2020.

Neal, R. M. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

Neal, R. M. and Hinton, G. E. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Packer, M. S. and Liu, D. R. Methods for the directed evolution of proteins. *Nature Reviews Genetics*, 16(7): 379–394, 2015.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Ren, Z., Li, J., Ding, F., Zhou, Y., Ma, J., and Peng, J. Proximal exploration for model-guided protein sequence design. *bioRxiv*, 2022.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised

learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Romero, P. A. and Arnold, F. H. Exploring protein fitness landscapes by directed evolution. *Nature reviews Molecular cell biology*, 10(12):866–876, 2009.

Sarkisyan, K. S., Bolotin, D. A., Meer, M. V., Usmanova, D. R., Mishin, A. S., Sharonov, G. V., Ivankov, D. N., Bozhanova, N. G., Baranov, M. S., Soylemez, O., et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016.

Seemayer, S., Gruber, M., and Söding, J. Ccmpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130, 2014.

Sinai, S., Wang, R., Whatley, A., Slocum, S., Locane, E., and Kelsic, E. D. Adalead: A simple and robust adaptive greedy search algorithm for sequence design. *arXiv preprint arXiv:2010.02141*, 2020.

Singh, A., Pandey, A., Srivastava, A. K., Tran, L.-S. P., and Pandey, G. K. Plant protein phosphatases 2c: from genomic diversity to functional multiplicity and importance in stress management. *Critical Reviews in Biotechnology*, 36(6):1023–1035, 2016.

Starita, L. M., Pruneda, J. N., Lo, R. S., Fowler, D. M., Kim, H. J., Hiatt, J. B., Shendure, J., Brzovic, P. S., Fields, S., and Klevit, R. E. Activity-enhancing mutations in an e3 ubiquitin ligase identified by high-throughput mutagenesis. *Proceedings of the National Academy of Sciences*, 110(14):E1263–E1272, 2013.

Starr, T. N. and Thornton, J. W. Exploring protein sequence–function landscapes. *Nature biotechnology*, 35(2):125–126, 2017.

Swersky, K., Rubanova, Y., Dohan, D., and Murphy, K. Amortized bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pp. 769–778. PMLR, 2020.

Terayama, K., Sumita, M., Tamura, R., and Tsuda, K. Black-box optimization for automated discovery. *Accounts of Chemical Research*, 54(6):1334–1346, 2021.

Trabucco, B., Geng, X., Kumar, A., and Levine, S. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pp. 21658–21676. PMLR, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Weile, J., Sun, S., Cote, A. G., Knapp, J., Verby, M., Mellor, J. C., Wu, Y., Pons, C., Wong, C., van Lieshout, N., et al. A framework for exhaustively mapping functional missense variants. *Molecular systems biology*, 13(12): 957, 2017.

Wrenbeck, E. E., Azouz, L. R., and Whitehead, T. A. Single-mutation fitness landscapes for an enzyme on multiple substrates reveal specificity is globally encoded. *Nature communications*, 8(1):1–10, 2017.

Wright, C. F., Teichmann, S. A., Clarke, J., and Dobson, C. M. The importance of sequence diversity in the aggregation and evolution of proteins. *Nature*, 438(7069): 878–881, 2005.

Zhang, D., Fu, J., Bengio, Y., and Courville, A. Unifying likelihood-free inference with black-box sequence design and beyond. *arXiv preprint arXiv:2110.03372*, 2021.

A. Data Statistics

We provide the detailed data statistics in the following table, including protein sequence length, data size and data source. We have checked and cleaned the data and make sure the data do not contain personally identifiable information or offensive content.

Protein	Length	Size	Data Source
avGFP	237	$49,855$	https://figshare.com/articles/dataset/Local_fitness_landscape_of_the_green_fluorescent_protein
AAV	28	$296,914$	https://github.com/churchlab/Deep_diversification_AAV
TEM	286	$17,238$	https://github.com/facebookresearch/esm/tree/main/examples/data
E4B	102	$91,033$	https://figshare.com/articles/dataset
AMIE	341	$6,631$	https://figshare.com/articles/dataset/Normalized_fitness_values_for_AmiE_selections/3505901/2
LGK	439	$8,069$	https://figshare.com/articles/dataset
Pab1	75	$36,522$	https://figshare.com/articles/dataset
UBE2I	159	$5,355$	http://dalai.mshri.on.ca/ jweile/projects/dmsData/

Table 5. Detailed statistics of the eight protein datasets.

B. More Implementation Details

B.1. Additional Experimental Settings

We apply the annealing schedule for the KL term during $P_\theta(x)$ training process following β-VAE (Higgins et al., 2016) to prevent posterior collapse. Specifically, the KL term coefficient starts from 0 and is gradually increased to 1.0 as training goes on. At each iteration in importance sampling based EM learning process, the number of samples from current $Q_\phi(x, z)$ is set to 10% of the training data size.

Following (Ren et al., 2022), We construct the oracle model $f(x)$ by adopting the features produced by ESM-1b (Rives et al., 2021) with dimension 1280 and finetuning an Attention1D decoder to predict the fitness values. Since Brookes et al. (2019) state that the results are insensitive when λ is set in the range $[50, 100]$-th percentile of the fitness scores in the training set, we set λ to 50-th percentile of the fitness values in the training data to accommodate more diversity.

B.2. Importance Weighted EM Learning Algorithm

Algorithm 1 Importance Sampling based Expectation-Maximization Training

Input: ε: separately learned combinatorial structure features through MRFs
$P_\theta(x;\varepsilon)$: VAE model trained on the protein sequences incorporating ε
T: number of iteration for importance sampling based EM learning
N: number of samples at each iteration during EM learning
Output: Final proposal model $Q_{\phi^{(T)}}(x;\varepsilon)$

```
set Q_{φ^(0)}(x|z;ε) = P_θ(x|z;ε),   Q_{φ^(0)}(z|x;ε) = P_θ(z|x;ε)
for t=0 to T-1 do
    sample N pairs of (x^(t), z^(t)) ~ Q_{φ^(t)}(x, z;ε)
    for minibatch in {(x^(t), z^(t))}_{i=1}^N do
        Calculate expectation using Monte Carlo approximation defined in Equation 13 in E-step
        Maximize the Monte Carlo approximation to update φ^(t+1) in Equation 14 in M-step
    end for
end for
```

C. Approximate KL Divergence

C.1. Proof of the Unbiased and Low-Variance Estimator

Letting $r(x) = \frac{P_\theta(x|\mathcal{S})}{Q_\phi(x)}$, we have:

$$E_{Q_\phi(x)}[(r(x)-1) - \log r(x)] = E_{Q_\phi(x)}[\log \frac{Q_\phi(x)}{P_\theta(x|\mathcal{S})}] = D_{KL}(Q_\phi(x)||P_\theta(x|\mathcal{S})) \tag{16}$$

Therefore, this estimator for KL divergence is unbiased.

Assuming $f(x) = (x-1) - \log x$, since $f(x)$ is a convex function and it achieves the minimum value when $x = 1$, we have:

$$(x-1) - \log x \geq f(1) = 0 \tag{17}$$

Thus, $(r(x)-1) - \log r(x)$ is always larger than or equals to 0. Instead, in the original KL divergence, $\log \frac{Q_\phi(x)}{P_\theta(x|\mathcal{S})} = -\log r(x)$ would be negative for half of the samples. Therefore, $E_{Q_\phi(x)}[(r(x)-1) - \log r(x)]$ has lower variance compared to the original one.

C.2. Theoretical Understanding

We can prove that under acceptable KL divergence, the samples from the proposal distribution $Q_\phi(x)$ can be bound within a reasonable sampling error with samples from the posterior distribution $P_\theta(x|\mathcal{S})$.

Lemma C.1. *If the KL divergence between two distributions P and Q is less than a small positive value δ, then the sampling probability difference between P and Q will be bounded by $\sqrt{2\delta}$ for each sample x.*

Proof. Let $\delta(P, Q)$ be the total variation distance between P and Q, due to the Pinsker's inequality (Csiszár & Körner, 2011), we have:

$$\frac{1}{2}\sum_x |P(x) - Q(x)| = \delta(P, Q) \leq \sqrt{\frac{1}{2}D_{KL}(P||Q)} < \sqrt{\frac{1}{2}\delta} \tag{18}$$

The first equality holds when the measurable space is discrete as is the case in this paper. The second inequality is tight if and only if $P = Q$, and then there is no difference between sampling from P and Q.

From the above analysis, we can get:

$$\sum_x |P(x) - Q(x)| < \sqrt{2\delta} \tag{19}$$

When δ approaches 0, the sampling difference between $P(x)$ and $Q(x)$ would be very minor.

D. Case Study

Figure 6 illustrates the complete sequence and secondary structure analyse of our designed protein of avGFP compared with Cytochrome b562 integral fusion with enhanced green fluorescent protein (EGFP). From the figure, we can see that there are much overlap between our designed protein sequence and the chain B of Cytochrome b562 integral fusion with EGFP. It gives empirical evidence that the green fluorescent protein generated by our model is highly likely to be a real protein compared with the proteins we already know. But whether the designed sequences can accelerate wet-lab experiments still need more exploration as we can not 100% trust it.

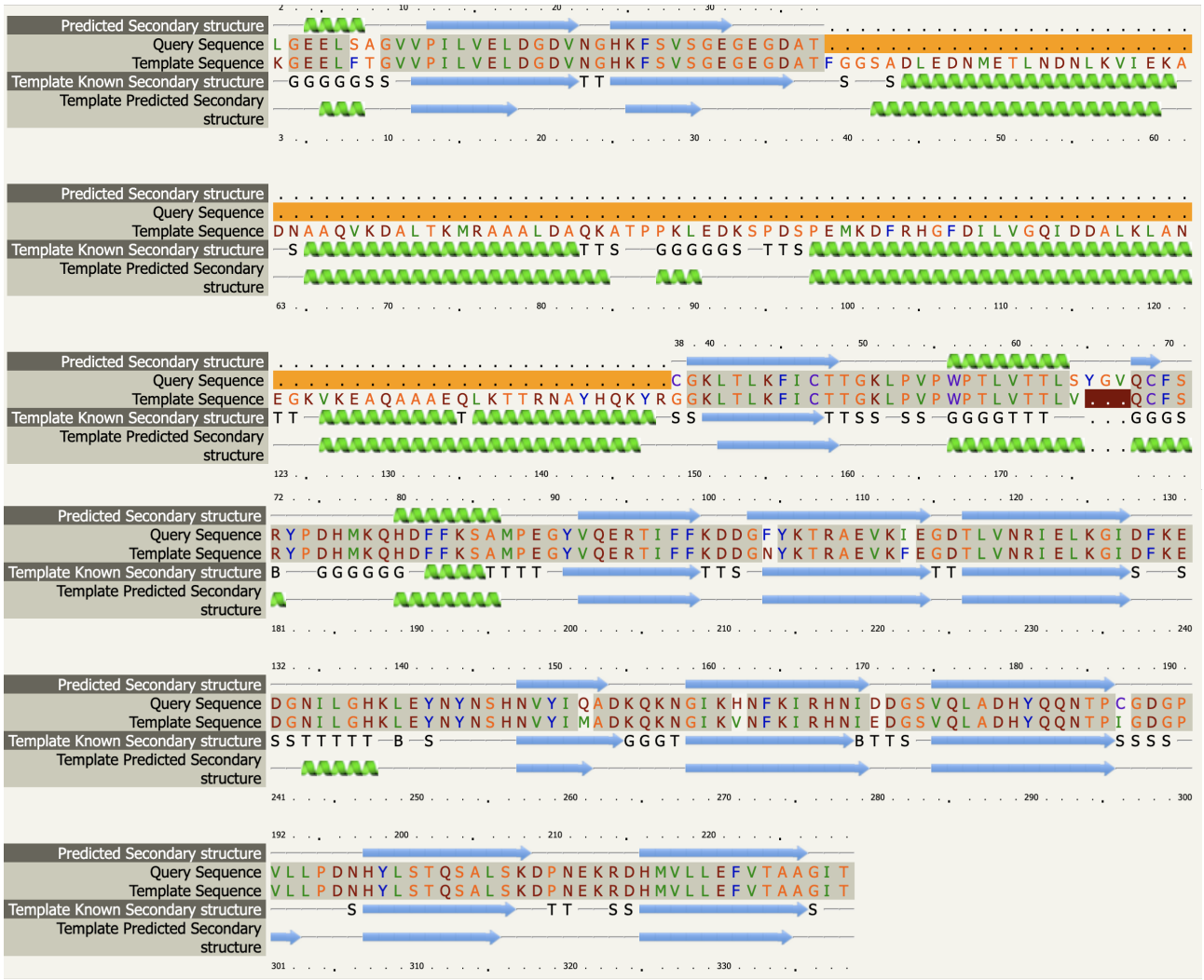


Figure 6. Complete sequence and secondary structure comparison between the designed sequence of avGFP and chain B of Cytochrome b562 integral fusion with with enhanced green fluorescent protein. Green and blue parts respectively represent the Alpha helix and Beta strand.

E. Pseudo-Likelihood for Combinatorial Structure Learning

We train the Markov random fields using a pseudo-likelihood as CCMpred (Seemayer et al., 2014) additionally combined with the Lasso regularization and Ridge regularization. The pseudo-likelihood is given in the following Equation:

$$
\begin{aligned}
\hat{P}_L(x) &= \log \Pi_{i=1}^{L} P(x_i | x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_L, \varepsilon) \\
&= \sum_{i=1}^{L} \log \frac{\exp(\varepsilon_i(x_i) + \sum_{j=1, j\neq i}^{L} \varepsilon_{ij}(x_i, x_j))}{\sum_{c\in\mathcal{V}} \exp(\varepsilon_i(c) + \sum_{j=1, j\neq i}^{L} \varepsilon_{ij}(c, x_j))} \\
&= \sum_{i=1}^{L} \{\varepsilon_i(x_i) + \sum_{j=1, j\neq i}^{L} \varepsilon_{ij}(x_i, x_j) - \log Z_i\} \\
Z_i &= \sum_{c\in\mathcal{V}} \exp(\varepsilon_i(c) + \sum_{j=1, j\neq i}^{L} \varepsilon_{ij}(c, x_j))
\end{aligned}
\tag{20}
$$

where $\mathcal{V}$ denotes the vocabulary of 20 amino acids.

F. Additional Experiments

F.1. Conditional VAE

We provide the results of conditional VAE (CVAE) in Table 6. We implement the CVAE which is built by adding the fitness condition on deepsequence (Riesselman et al., 2018). The table shows CVAE exhibits decreased performance. It demonstrates that iterative sampling inside EM algorithm with explicit fitness constraints can lead to better proteins.

Dataset	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	Average	
CVAE	3.570	4.128	0.083	−0.819	−1.564	−0.796	1.638	3.492	1.217
IsEM-Pro	6.185	4.813	1.850	5.737	0.062	0.035	2.923	4.536	**3.267**

Table 6. Fitness of conditional VAE compared with our IsEM-Pro.

F.2. Non-Autoregressive Decoder

We provide the performance of a fully-connected decoder implemented by applying a non-autoregressive decoder compared with our IsEM-Pro in Table 7. It shows the non-autoregressive decoder can not generate good candidates for protein with longer sequences such as LGK (439 amino acids).

Dataset	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	Average	
Non-autoregressive Decoder	1.312	3.137	1.015	3.096	−3.297	−3.125	1.548	2.694	0.797
IsEM-Pro	6.185	4.813	1.850	5.737	0.062	0.035	2.923	4.536	**3.267**

Table 7. Fitness of applying a fully-connected decoder (Non-autoregressive decoder) compared with our IsEM-Pro.