Bridge-IF: Learning Inverse Protein Folding with Markov Bridges

Yiheng Zhu[1], **Jialu Wu**[2], **Qiuyi Li**[3], **Jiahuan Yan**[1], **Mingze Yin**[4], **Wei Wu**[5], **Mingyang Li**[3], **Jieping Ye**[3], **Zheng Wang**[3]*, **Jian Wu**[1,4,6*]
[1]College of Computer Science & Technology and Liangzhu Laboratory, Zhejiang University
[2]College of Pharmaceutical Sciences, Zhejiang University
[3]Alibaba Cloud Computing
[4]School of Public Health, Zhejiang University
[5]School of Artificial Intelligence and Data Science, University of Science and Technology of China
[6]The Second Affiliated Hospital Zhejiang University School of Medicine
{zhuyiheng2020, jialuwu, jyansir, yinmingze, wujian2000}@zju.edu.cn
{liqiuyi.lqy, sangheng.lmy, yejieping.ye, wz388779}@alibaba-inc.com
urara@mail.ustc.edu.cn

Abstract

Inverse protein folding is a fundamental task in computational protein design, which aims to design protein sequences that fold into the desired backbone structures. While the development of machine learning algorithms for this task has seen significant success, the prevailing approaches, which predominantly employ a discriminative formulation, frequently encounter the error accumulation issue and often fail to capture the extensive variety of plausible sequences. To fill these gaps, we propose Bridge-IF, a generative diffusion bridge model for inverse folding, which is designed to learn the probabilistic dependency between the distributions of backbone structures and protein sequences. Specifically, we harness an expressive structure encoder to propose a discrete, informative prior derived from structures, and establish a Markov bridge to connect this prior with native sequences. During the inference stage, Bridge-IF progressively refines the prior sequence, culminating in a more plausible design. Moreover, we introduce a reparameterization perspective on Markov bridge models, from which we derive a simplified loss function that facilitates more effective training. We also modulate protein language models (PLMs) with structural conditions to precisely approximate the Markov bridge process, thereby significantly enhancing generation performance while maintaining parameter-efficient training. Extensive experiments on well-established benchmarks demonstrate that Bridge-IF predominantly surpasses existing baselines in sequence recovery and excels in the design of plausible proteins with high foldability. The code is available at `https://github.com/violet-sto/Bridge-IF`.

1 Introduction

Proteins are 3D folded linear chains of amino acids that execute the myriad of biological processes fundamental to life, such as catalysing metabolic reactions, mediating immune responses, and responding to stimuli [23]. Designing protein sequences that fold into desired 3D structures, known as inverse protein folding, is a crucial task with great potential for applications in protein engineering [29, 66, 5]. Beyond long-established physics-based methods like Rosetta [2], the considerable promise of leveraging geometric deep learning for protein structure modeling has given rise to an ongoing

*Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1260

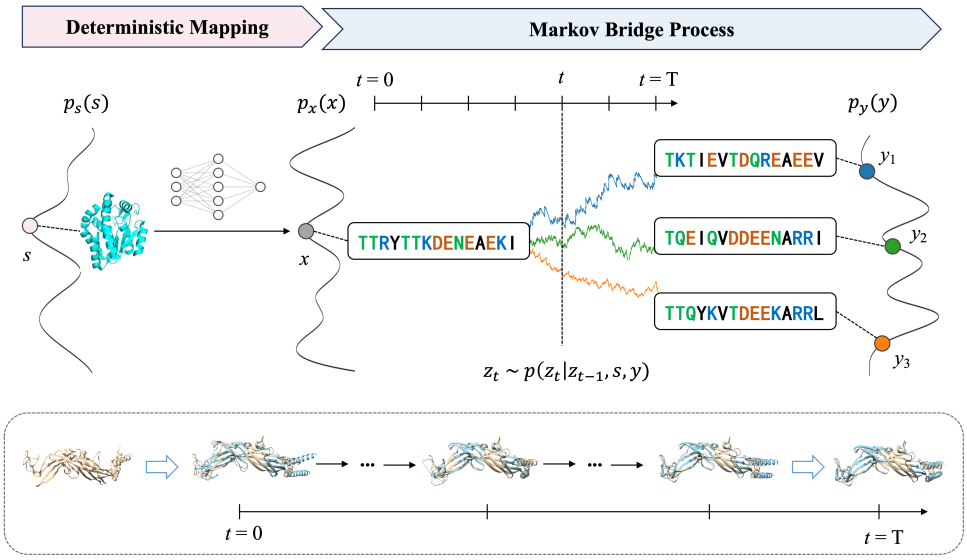

Figure 1: **Overview of Bridge-IF**. Bridge-IF consists of an expressive structure encoder supervised by native sequences for proposing a discrete, deterministic prior, and a Markov bridge model for learning the dependency between the distribution of prior sequences and the distribution of native sequences. During the inference stage, Bridge-IF progressively refines the prior sequence.

paradigm. This paradigm is centered on deciphering the principles of protein design directly from data and on predicting sequences corresponding to specific structures [25, 27, 6, 22].

Despite substantial advancements, most existing approaches follow a discriminative formulation for learning inverse folding [58], consequently encountering two principal obstacles: (i) *Error accumulation issue*. For instance, Transformer-based autoregressive models are constrained by their inherent sequential generation process and exposure bias, which prevents them from correcting preceding erroneous predictions. (ii) *One-to-many mapping nature of the inverse folding problem.* A multitude of distinct amino acid sequences possess the capability to fold into an identical protein backbone structure, a phenomenon exemplified by homologous proteins. Discriminative models are incapable of capturing the one-to-many mapping from the protein structure to non-unique sequences, thereby facing difficulties in covering the broad spectrum of plausible solutions [58].

Recent studies have advanced the iterative refinement strategy to optimize the previously generated results, aiming to reduce prediction errors [64, 14, 42]. These approaches employ a refinement module to identify and correct inaccurately predicted amino acids. However, as the number of refinement iterations grows, managing the intermediate stages effectively becomes more challenging, potentially hindering sustained performance gains.

Diffusion-based generative models [45, 17], particularly their discrete extensions [3], which offer a structured iterative refinement process with probabilistic interpretation, appear to be a promising solution. GraDe-IF [58] is a pioneer in investigating diffusion models for inverse folding, leveraging the backbone structure to guide the denoising process on the amino acid residues. However, as diffusion models are designed to learn a single intractable data distribution, the prior distribution utilized by GraDe-IF is restricted to a simple noise distribution (i.e., a uniform distribution across all residue types), which has little or no information about the distribution of native sequences. It remains unclear whether this default formulation best suits conditional generative problems such as inverse protein folding, where the backbone structures provide significantly more information than random noise. Thus, an exciting research question naturally arises: *Can we propose a more strong and informative prior based on desired backbone structures to enhance the quality of samples and accelerate the inference process?*

In this work, we propose Bridge-IF, a novel generative diffusion bridge model for inverse folding. Its core design is aimed at generating protein sequences from a structure-aware prior. As shown

in Figure 1, we leverage an expressive structure encoder supervised by native sequences to propose a discrete, deterministic prior based on desired structures, and build a Markov bridge [10, 24] between it and the native sequence. By approximating the reference Markov bridge process, Bridge-IF learns to progressively refine the prior sequence, resulting in a more plausible design. Furthermore, we present a fresh reparameterization perspective on Markov bridge models and derive a simplified loss function that yields enhanced training effectiveness. Inspired by significant advances in protein language models (PLMs) for understanding proteins [9, 34], we innovatively integrate conditions, including timestep and structures, into PLMs to accurately approximate the Markov bridge process. This approach notably improves generation performance while ensuring parameter-efficient training. Empirically, we demonstrate that Bridge-IF outperforms state-of-the-art baselines on several standard benchmarks and excels in the design of plausible proteins with high foldability.

To summarise, the main contributions of this work are as follows:

- We introduce Bridge-IF, the first generative diffusion bridge model based on Markov bridges for inverse folding. We also offer a reparameterization perspective and derive a simplified loss function to facilitate effective training.
- We innovatively adapt PLMs to effectively capture both timestep and structural information while ensuring the modified architecture is compatible with pre-trained weights.
- Experiments verify that Bridge-IF achieves state-of-the-art performance on standard benchmarks.

2 Related work

2.1 Inverse protein folding

Recently, AI algorithms have spurred a revolution in modeling protein folding [28, 34]. Meanwhile, the inverse problem of protein folding, which aims to infer an amino acid sequence that will fold into the desired structure, is gaining increasing attention [6]. By representing protein backbone structures as a k-NN graph, geometric deep learning has achieved remarkable progress in learning inverse folding [25, 6, 22], surpassing traditional physics-based approaches [2], and even facilitating the design of a range of experimentally validated proteins [6, 56]. Modern deep learning-based inverse folding approaches typically comprise a structure encoder and a sequence decoder. Depending on their decoding strategies, these approaches can be classified into three categories: autoregressive models, one-shot models, and iterative models. Most methods adopt the autoregressive decoding scheme to generate amino acid sequences [25, 6, 22]. Given that autoregressive models tend to have low inference speed, some researchers have investigated one-shot methods that facilitate the parallel generation of multiple tokens [12, 38]. Since directly predicting highly plausible sequences is challenging, some works have shifted their attention to iterative refinement [64, 14, 26, 42, 58]. For instance, LM-Design [64] and KW-design [14] utilize the pre-trained knowledge from PLMs to reconstruct a native sequence from a corrupted version. The Potts model-based ChromaDesign [26] and CarbonDesign [42] employ iterative sampling techniques, including Markov chain Monte Carlo, to design protein sequences. GraDe-IF [58] further leverages the principles of discrete denoising diffusion probabilistic models [3], demonstrating a strong capacity to encompass diverse plausible solutions. In this work, we present the first generative diffusion bridge model for inverse folding.

2.2 Diffusion models

Diffusion-based generative models [45, 17] have showcased remarkable successes in a wide range of applications, ranging from image synthesis [8], audio synthesis [31], to video generation [18]. Generally, the essential idea behind these models is to define a forward diffusion process that gradually transforms the data into a simple prior distribution and learn a reverse denoising process to gradually recover original data samples from the prior distribution. While most existing methods are designed for modeling continuous data, a few efforts have extended diffusion models to discrete data domains [3, 33, 52, 36]. Recently, diffusion models have also found utility in scientific discovery [55], particularly in protein design [56, 59, 1, 15, 58].

 https://doi.org/10.52202/079017-1260

2.3 Schrödinger bridge problem

The Schrödinger bridge (SB) problem is a classical entropy-regularized optimal transport problem [43, 32, 4]. Given a data distribution, a prior distribution, and a reference stochastic process between them, solving the SB problem amounts to finding the closest process to the reference in terms of Kullback-Leibler divergence on path spaces. This concept exhibits fundamental similarities to diffusion models [47], particularly in the field of unconditional generative modeling [49, 54, 7, 44], where the prior distribution assumes the form of Gaussian noise. Notably, SB formalism offers a general framework for approximating the reference stochastic process by training on coupled samples from two continuous distributions [19, 46, 35, 65]. The recently proposed Markov bridge [10, 24] has broadened the scope of the SB, enabling it to model categorical distributions. In this work, we present the first diffusion bridge model for inverse protein folding.

3 Background

3.1 Problem formulation and notation

Generally, a protein can be represented as a pair of amino acid sequence and structure $(\boldsymbol{y}, \boldsymbol{s})$, where $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$ denotes its sequence of n residues with $y_i \in \{1, 2, \ldots, 20\}$ indicating the type of the i-th residue, and $\boldsymbol{s} = [s_1, s_2, \ldots, s_n] \in \mathbb{R}^{n\times 4\times 3}$ denotes its structure with s_i representing the Cartesian coordinates of the i-th residue's backbone atoms (i.e., N, C-α, and C, with O optionally). The inverse protein folding problem aims to automatically identify the protein sequence $\boldsymbol{y}$ that can fold into the given structure $\boldsymbol{s}$. Given that homologous proteins invariably exhibit similar structures, the solution for a given structure is not unique [16]. Hence, an ideal model, parameterized by θ, should be capable of learning the underlying mapping from protein backbone structures to their corresponding sequence distributions $p_\theta(\boldsymbol{y}|\boldsymbol{s})$.

3.2 Markov bridge models

Markov bridge model [24] is a general framework for learning the probabilistic dependency between two intractable discrete-valued distributions $p_\mathcal{X}$ and $p_\mathcal{Y}$. For a pair of samples $(\boldsymbol{x}, \boldsymbol{y}) \sim p_{\mathcal{X},\mathcal{Y}}(\boldsymbol{x}, \boldsymbol{y})$, it defines a Markov process pinned to fixed start and end points $\boldsymbol{z}_0 = \boldsymbol{x}$ and $\boldsymbol{z}_T = \boldsymbol{y}$ through a sequence of random variables $(\boldsymbol{z}_t)_{t=0}^T$ that satisfies the Markov property,

$$p(\boldsymbol{z}_t|\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}, \boldsymbol{y}) = p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{y}). \tag{1}$$

To pin the process at the end point $\boldsymbol{z}_T = \boldsymbol{y}$, we have an additional requirement,

$$p(\boldsymbol{z}_T = \boldsymbol{y}|\boldsymbol{z}_{T-1}, \boldsymbol{y}) = 1. \tag{2}$$

Assuming that both $p_\mathcal{X}$ and $p_\mathcal{Y}$ are categorical distributions with a finite sample space $\{1, \ldots, K\}$, we can represent data points as one-hot vectors: $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_t \in \{0, 1\}^K$, and define the transition probabilities (Equation 1) as follows,

$$p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y}) = \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t\boldsymbol{z}_t\right), \tag{3}$$

where $\text{Cat}(\cdot\,; \boldsymbol{p})$ is a categorical distribution with probabilities given by $\boldsymbol{p}$, and $\boldsymbol{Q}_t$ is a transition matrix parameterized as

$$\boldsymbol{Q}_t \coloneqq \boldsymbol{Q}_t(\boldsymbol{y}) = \beta_t\boldsymbol{I}_K + (1 - \beta_t)\boldsymbol{y}\boldsymbol{1}_K^\top, \tag{4}$$

where β_t is a schedule parameter transitioning from $\beta_0 = 1$ to $\beta_{T-1} = 0$. It is easy to see that $\boldsymbol{z}_t$ can be efficiently sampled from $p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_0, \boldsymbol{z}_T) = \text{Cat}\left(\boldsymbol{z}_{t+1}; \overline{\boldsymbol{Q}}_t\boldsymbol{z}_0\right)$ with a cumulative product matrix $\overline{\boldsymbol{Q}}_t = \boldsymbol{Q}_t\boldsymbol{Q}_{t-1}...\boldsymbol{Q}_0 = \overline{\beta}_t\boldsymbol{I}_K + (1 - \overline{\beta}_t)\boldsymbol{y}\boldsymbol{1}_K^\top$, where $\overline{\beta}_t = \prod_{s=0}^t \beta_s$.

Training Using the finite set of coupled samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^D \sim p_{\mathcal{X},\mathcal{Y}}$, Markov bridge model learns to sample $\boldsymbol{y}$ when only $\boldsymbol{x}$ is available by approximating $\boldsymbol{y}$ with a neural network φ_θ:

$$\hat{\boldsymbol{y}} = \varphi_\theta(\boldsymbol{z}_t, t), \tag{5}$$

and defining an approximated transition kernel,

$$q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) = \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\hat{\boldsymbol{y}})\boldsymbol{z}_t\right). \tag{6}$$

φ_θ is trained by optimizing the variational bound on negative log-likelihood $\log q_\theta(\boldsymbol{y}|\boldsymbol{x})$, which has the following closed-form expression,

$$-\log q_\theta(\boldsymbol{y}|\boldsymbol{x}) \le T \cdot \mathbb{E}_{t\sim\mathcal{U}(0,\ldots,T-1)} \underbrace{\mathbb{E}_{\boldsymbol{z}_t\sim p(\boldsymbol{z}_t|\boldsymbol{x},\boldsymbol{y})} D_{\text{KL}}\left(p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y}) \| q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)\right)}_{\mathcal{L}_t}. \tag{7}$$

Sampling To sample a data point $\boldsymbol{y} \equiv \boldsymbol{z}_T$ starting from a given $\boldsymbol{z}_0 \equiv \boldsymbol{x} \sim p_{\mathcal{X}}(\boldsymbol{x})$, one can iteratively predict $\hat{\boldsymbol{y}} = \varphi_\theta(\boldsymbol{z}_t, t)$ and then derive $\boldsymbol{z}_{t+1} \sim q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) = \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\hat{\boldsymbol{y}})\boldsymbol{z}_t\right)$ for $t = 0, \ldots, T-1$.

4 Methods

In this section, we introduce Bridge-IF, a Markov bridge-based model for inverse protein folding. Figure 1 shows an overview of our proposed Bridge-IF. Due to space limitation, we present the detailed algorithm in Appendix A. To begin, we describe how to extend Markov bridge techniques to facilitate the inverse protein folding task. Next, we propose a simplified training objective. Finally, we elucidate how to modulate pre-trained PLMs with structural conditions to approximate the Markov bridge process.

4.1 Overview of Bridge-IF

We frame the inverse protein folding problem as a generative problem of modeling a stochastic process between the distributions of backbone structures $p_{\mathcal{S}}(s)$ and protein sequences $p_{\mathcal{Y}}(y)$. As previously discussed, diffusion bridge models, with their general properties of an unrestricted prior form, serves as an ideal substitution for diffusion models in the presence of a well-defined informative prior. Regrettably, to the best of our knowledge, no existing method can directly model the dependency between two distinct types of distributions: specifically, the *continuous* source distribution of backbone structures and the *discrete* target distribution of protein sequences.

To reconcile the differences between source and target distributions and streamline the modeling process, we propose introducing a discrete proposal distribution to serve as a deterministic prior. We parameterize the proposal distribution using a structure encoder $\mathcal{E} : \mathcal{S} \to \mathcal{X}$ that is supervised by ground-truth target sequences. Recent advancements have demonstrated that an expressive encoder is capable of directly predicting pretty good protein sequences in a one-shot manner [12]. This approach enables us to utilize structural information more effectively, rather than simply employing it to guide the denoising process as in previous diffusion-based methods like GraDe-IF [58]. In this work, we will take the discriminative model PiFold [12] as the structure encoder to produce a clean and deterministic prior $\boldsymbol{x} = \mathcal{E}(\boldsymbol{s})$. Upon this deterministic mapping from structure to sequence, we simplify the originally complex problem of modeling $p(\boldsymbol{s}, \boldsymbol{y})$ into the more tractable problem of modeling $p(\boldsymbol{x}, \boldsymbol{y})$. Then, we build a Markov bridge [10, 24] between the prior sequence and the native sequence to model the stochastic process, leading to a data-to-data process. As depicted in the lower half of Figure 1, each sampling step progressively refines the prior sequence, which contains significant information about the target sequence, ultimately resulting in a more precise prediction.

Recall that the Markov bridge models are typically trained by optimizing the variational bound on negative log-likelihood $\log q_\theta(\boldsymbol{y}|\boldsymbol{x})$ (Equation 7), which is analytically complicated and hard to optimize in practice [63, 62]. Therefore, we here propose a reparameterization perspective on Markov bridge models, deriving a simplified loss function for easier optimization (§4.2).

We build Markov bridges in the sequence space, treating the sequence representation as a set of independent categorical random variables. To model the Markov bridge process, $\boldsymbol{Q}_t$ is applied separately to each residue within a protein sequence. Motivated by the impressive advancements in PLMs for understanding and generating proteins [9, 34, 37, 60], we advocate for employing PLMs to approximate the Markov bridge process. This approach capitalizes on the emergent evolutionary knowledge of proteins, learned from an extensive dataset of protein sequences. Additionally, we utilize the latent structural features extracted by the structure encoder to prompt PLMs, thereby guiding the generation of structurally coherent proteins. Formally, the final state of the Markov bridge process is approximated by $\hat{\boldsymbol{y}} = \varphi_\theta(\boldsymbol{z}_t, \boldsymbol{s}, t)$, foregoing the use of Equation 5. We investigate the integration of conditional information, such as timestep and structure, into PLMs, focusing on preserving their emergent knowledge and achieving parameter-efficient training (§4.3).

4.2 Reparameterized Markov bridge models

Inspired by the similarities between Markov bridge models [24] and discrete diffusion models [3, 63], we propose a reparameterization of the Markov bridge model characterized in §3.2 to enable more effective training. With the reparameterization trick, we introduce a latent binary random variable

https://doi.org/10.52202/079017-1260

$v_t \sim \text{Bernoulli}(\overline{\beta}_{t-1})$ to indicate whether $\boldsymbol{z}_t$ has been transformed from $\boldsymbol{z}_0$ to $\boldsymbol{z}_T$. Thus $\boldsymbol{z}_t$ can be sampled from $p(\boldsymbol{z}_t|v_t, \boldsymbol{z}_0, \boldsymbol{y}) = v_t\boldsymbol{z}_0 + (1 - v_t)\boldsymbol{y}$. Accordingly, $p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y})$ can be equivalently written as:

$$p(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t, \boldsymbol{y}) = \begin{cases} \boldsymbol{z}_t & \text{if } v_t = 0 \\ (1 - \beta_t)\boldsymbol{y} + \beta_t \boldsymbol{z}_t & \text{if } v_t = 1 \end{cases} \tag{8}$$

Using the teacher-forcing approach, we can similarly define the approximation process as

$$q_\theta(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t) = \begin{cases} \boldsymbol{z}_t & \text{if } v_t = 0 \\ (1 - \beta_t)\varphi_\theta(\boldsymbol{z}_t, t) + \beta_t \boldsymbol{z}_t & \text{if } v_t = 1 \end{cases} \tag{9}$$

Proposition 4.1. *The loss objective $\mathcal{L}_t(\theta)$ for sequence x at the t-th step can be reduced to the form*

$$\mathcal{L}_t(\theta) = \lambda_t \mathbb{E}_{p(\boldsymbol{z}_t|\boldsymbol{x},\boldsymbol{y})}[-v_t \boldsymbol{y}^T \log \varphi_\theta(\boldsymbol{z}_t, t)], \tag{10}$$

where $\lambda_t = 1 - \beta_t$.

The full derivation is provided in C. This derived expression of $\mathcal{L}_t(\theta)$ formulates the training loss as a re-weighted standard multi-class cross-entropy loss function, which is computed over tokens that have not been transformed to the ground truth $\boldsymbol{y} = \boldsymbol{z}_T$. Following Ho et al. [17], we set λ_t to a constant 1 in practice. Compared to the simpler cross-entropy loss calculated across all tokens, this new formulation places greater weight on tokens that require refinement. On the other hand, it is conceptually simpler than the original training loss (Equation 7), which requires calculating the complicated KL divergence between two categorical distributions $D_{\text{KL}}[p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y})\|q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)]$.

4.3 Network architecture design space

We adopt pre-trained PLMs as the base network to approximate the final state of the Markov bridge process. Typically, PLMs exclusively take protein sequences as input during the pre-training stage, making it non-trivial to integrate timestep and structural conditions into the PLMs. Hence, we innovatively tailor the Transformer blocks [50] to effectively capture timestep and structural information, as depicted in Figure 2. To facilitate efficient training, the architecture of our model is delicately designed for compatibility with the pre-trained weights. Our exposition emphasizes fundamental principles and the corresponding modifications to the base network.

4.3.1 AdaLN-Bias

Inspired by DiT [41], we explore replacing standard layer norm layers in transformer blocks with adaptive layer norm (adaLN) to modulate the normalization's output based on both the timestep of the Markov bridge process and the backbone structure. The key idea is to regress the dimension-wise scale and shift parameters γ and β of the layer norm from the sum of the timestep embedding and the pooled structure representation. In our situation, meaningful pre-trained parameters γ and β are readily accessible. Upon commencing the fine-tuning stage, it is crucial that these parameters are close to the pre-trained values to preserve the effectiveness of the original model, since a poor initialization could significantly deteriorate performance. For simplicity, we propose to predict bias $\Delta\gamma$ and $\Delta\beta$ on the frozen original scalars and initialize the multi-layer perception (MLP) to output the zero-vector for all $\Delta\gamma$ and $\Delta\beta$. We term the proposed variant of adaLN as adaLN-Bias.

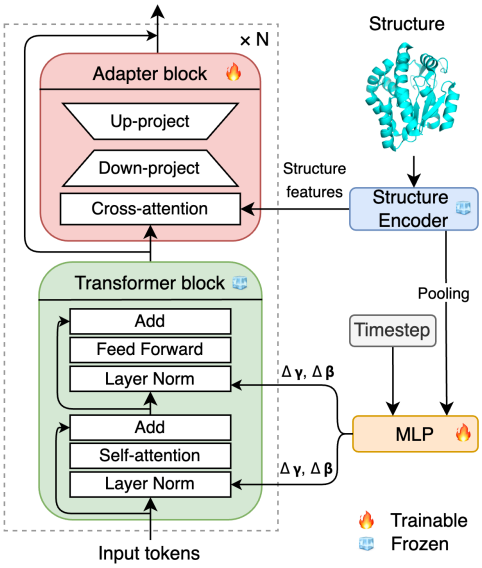

Figure 2: Model architecture of Bridge-IF.

Table 1: Results comparison on the **CATH** dataset. Benchmarked results are quoted from Hsu et al. [22], Zheng et al. [64], Yi et al. [58], Gao et al. [14]. †: "Single-chain" in Hsu et al. [22] is defined differently. The **best** and suboptimal results are labeled with bold and underline.

	Model	Perplexity ↓			Recovery Rate % ↑		
		Short	Single-chain	All	Short	Single-chain	All
CATH v4.2	StructGNN [25]	8.29	8.74	6.40	29.44	28.26	35.91
	GraphTrans [25]	8.39	8.83	6.63	28.14	28.46	35.82
	GCA [48]	7.09	7.49	6.05	32.62	31.10	37.64
	GVP [27]	7.23	7.84	5.36	30.60	28.95	39.47
	AlphaDesign [11]	7.32	7.63	6.30	34.16	32.66	41.31
	ProteinMPNN [6]	6.21	6.68	4.61	36.35	34.43	45.96
	PiFold [12]	6.04	6.31	4.55	39.84	38.53	51.66
	GraDe-IF [58]	**5.49**	6.21	4.35	**45.27**	42.77	52.21
	With PLMs						
	LM-Design (ESM-1b 650M) [64]	6.77	6.46	4.52	37.88	42.47	55.65
	KW-Design (ESM-2 650M) [14]	6.05	5.29	3.90	43.32	46.30	57.38
	Bridge-IF (ESM-1b 650M)	5.67	**5.27**	**3.90**	43.84	**48.24**	**58.49**
	Bridge-IF (ESM-2 650M)	5.68	**5.06**	**3.83**	43.86	**48.96**	**58.59**
CATH v4.3	GVP-large [22]	7.68	6.12†	6.17	32.60	39.40†	39.20
	ESM-IF [22]	8.18	6.33†	6.44	31.30	38.50†	38.30
	+1.2M AF2 predicted data [22]	6.05	4.00†	4.01	38.10	51.50†	51.60
	With PLMs						
	LM-Design (ESM-1b 650M) [64]	5.66	5.52	4.01	46.84	48.63	56.63
	Bridge-IF (ESM-1b 650M)	**5.17**	**4.63**	**3.68**	**50.00**	**53.49**	**58.93**

4.3.2 Structural adapter

Considering that the pooled structure representation might only retain coarse-grained information, the network could consequently lack a detailed understanding of the structure input and necessitate information derived from original structural features to compensate. We incorporate a multi-head cross-attention module to the transformer block, enabling the network to flexibly interact with the structural features extracted from the structure encoder [64]. To facilitate pre-trained weights, we further integrate it into a bottleneck adapter layer [21] with residual connection, preserving the input for the subsequent layers.

We stress that we freeze all pre-trained parameters of the base network during training.

5 Experiments

In this section, we first demonstrate the effectiveness of our Bridge-IF on the standard CATH benchmark [40]. Next, we assess Bridge-IF for its applicability in *de novo* protein design. Moreover, we conduct several ablation studies to empirically justify the key design choices. Further results pertaining to the design of multi-chain protein complexes can be found in Appendix B.1.

5.1 Experimental protocol

Training setup We conduct experiments on both **CATH v4.2** and **CATH v4.3**, where proteins are categorized based on the CATH hierarchical classification of protein structure, to ensure a comprehensive analysis. Following the standard data splitting provided by Ingraham et al. [25], CATH v4.2 dataset consists of 18,024 proteins for training, 608 proteins for validation, and 1,120 proteins for testing. Following the standard data splitting provided by Hsu et al. [22], CATH v4.3 dataset consists of 16,153 proteins for training, 1,457 proteins for validation, and 1,797 proteins for testing. For a fair comparison with iterative models [64, 14], we use pre-trained PiFold [12] to propose the prior distribution. We use the cosine schedule [39] with number of timestep $T = 25$. The model is trained up to 50 epochs by default on an NVIDIA 3090. We used the same training

https://doi.org/10.52202/079017-1260

settings as ProteinMPNN [6], where the batch size was set to approximately 6000 residues, and Adam optimizer [30] with noam learning rate scheduler [51] was used.

Baselines We compare Bridge-IF with several state-of-the-art baselines, categorized into three groups: (1) autoregressive models, including StructGNN [25], GraphTrans [25], GCA [48], GVP [27], AlphaDesign [11], ESM-IF [22], and ProteinMPNN [6]; (2) the one-shot model, PiFold [12]; (3) iterative models, including LM-Design [64], KW-Design [14], and diffusion-based GraDe-IF [58].

Evaluation We evaluate the generative quality using *perplexity* and *recovery rate*. Following previous studies [25, 22], we report perplexity and median recovery rate on three settings, namely short proteins (length ≤ 100), single-chain proteins (labeled with 1 chain in CATH), and all proteins.

5.2 Inverse folding

The performance of Bridge-IF, compared to competitive baselines, is summarized in Table 1. Bridge-IF demonstrates superior performance over previous methods. We highlight the following: (1) Iterative models comprehensively surpass the previously dominant autoregressive and one-shot methods. (2) Our Bridge-IF outperforms LM-Design and KW-Design with the same pre-trained PLMs, supporting our hypothesis that the iterative refinement process should be modeled in a probabilistic framework. (3) Compared with diffusion-based GraDe-IF, our Bridge-IF achieves better performance with fewer diffusion steps (25 vs. 500), demonstrating that our bridge-based formulation can better leverage the structural prior.

Following Zheng et al. [64], we also study the impact of the scale of PLMs on CATH v4.3. We use ESM-2 series, with parameters ranging from 8M to 3B. As depicted in Figure 3, the performance of Bridge-IF improves with model scaling, exhibiting a distinct scaling law in logarithmic scale. Using ESM-2 at the same scale, we observe that Bridge-IF consistently obtains greater enhancements relative to LM-Design. Besides, Bridge-IF does not exhibit any performance degradation, even when the smallest model (i.e, ESM-2 8M) is employed. Remarkably, the largest ESM2-3B-based variant of Bridge-IF attains a record-setting recovery rate of 61.27% on CATH v4.3.

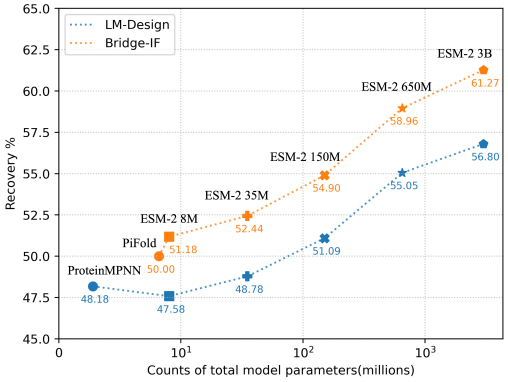

Figure 3: Performance comparison w.r.t. model scales of pLMs using ESM-2 series on CATH 4.3.

5.3 Foldability

While perplexity and recovery rate serve as effective proxy metrics, it is imperative to recognize that these measurements may not accurately reflect the foldability of the designed protein sequences in real-world scenarios [58, 13, 53]. Given that wet-lab assessment is extremely costly, we leverage the *in silico* structure prediction model ESMFold [34], to evaluate whether our designs can adhere to the structure condition. Here we assess the agreement of the native structures with the predicted structures using the TM-score [61], and follow the evaluation configurations as in Wang et al. [53]. Specifically, we use the small, high-quality test set of 82 samples curated by Wang et al. [53] and randomly generate 100 sequences for each structure.

Table 2: Numerical comparison on foldability and recovery rate. Benchmarked results are quoted from Wang et al. [53]. The **best** and suboptimal results are labeled with bold and underline.

Model	TM-score	Recovery %
Native sequences	0.80	100.00
Uniform	0.05	5.00
Natural frequencies	0.07	5.84
GraphTrans	0.72	35.89
GVP	0.73	39.46
ProteinMPNN	<u>0.80</u>	41.44
PiFold	0.71	44.86
LM-Design	0.73	<u>51.23</u>
Bridge-IF	**0.81**	**54.08**

We report the TM-score and recovery metrics in Table 2. We observe that our Bridge-IF stands out as the leading model, exhibiting both high

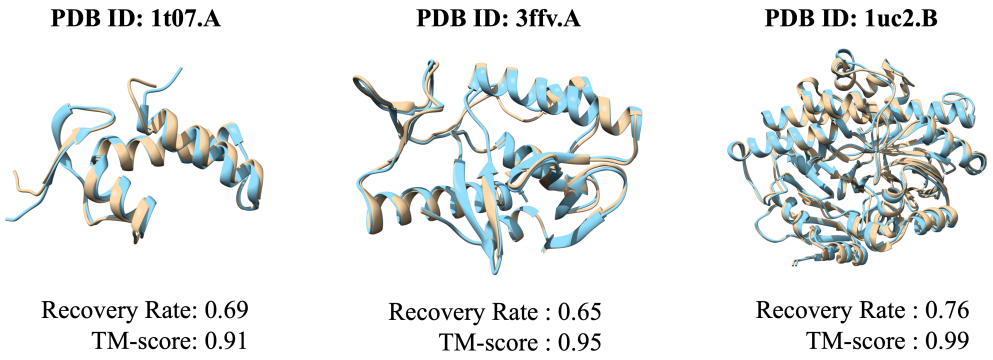

Figure 4: Folding comparison of our designed sequences (in blue) and the native sequences (in nude).

Table 3: Ablation studies of key design choices on CATH v4.2. "w/ AdaLN-Bias" replaces the vanilla AdaLN with AdaLN-Bias. "w/ SCE" replaces the variational lower bound loss with simplified cross-entropy loss.

Prior	**Architecture**	**Objective**	**Perplexity** ↓			**Recovery Rate** % ↑		
w/ pre-training	w/ AdaLN-Bias	w/ SCE	Short	Single-chain	All	Short	Single-chain	All
	✓	✓	6.51	6.30	4.23	43.17	44.29	56.53
✓		✓	5.98	5.27	3.89	43.45	48.01	57.92
✓	✓		6.52	6.40	4.28	43.43	44.01	56.43
✓	✓	✓	**5.68**	**5.06**	**3.83**	**43.86**	**48.96**	**58.59**

foldability and a high recovery rate. Notably, the predicted structures of our redesigned sequences align more closely with the given structures than do the native sequences, implying better structural validity of our redesigns. Another interesting finding is that PiFold and LM-Design achieve high recovery via a discriminative formulation but fall short on TM-score, indicating the limitation of structure-agnostic metrics. In contrast, probabilistic models Bridge-IF and ProteinMPNN,[2] perform exceptionally well on foldability. These results support our hypothesis that inverse protein folding should be modeled in a probabilistic framework considering the absence of a unique native sequence for a given backbone structure. Figure 4 showcases several instances where the folded structures of sequences designed by Bridge-IF are compared with reference crystal structures.

5.4 *De novo* protein design

Thus far, our experiments have been limited to accurate experimentally-determined structures. However, in real-world applications like *de novo* protein design, inverse folding models are commonly used to design sequences for novel structures generated by backbone generation models [56, 26]. Consequently, we next evaluate Bridge-IF for its potential in such a scenario. The experimental methodology is detailed as follows: we sample 10 backbones at every length $[100, 105, \ldots, 500]$ in intervals of 5 using Chroma [26]. For each de novo structure, we employ inverse folding models to design 8 sequences. Subsequently, these sequences are folded using ESMFold to identify the sequence with the highest TM-score (scTM). We compare Bridge-IF with ProteinMPNN [6], which is widely used in *de novo* protein design [59, 57]. Our results show that Bridge-IF surpasses ProteinMPNN in terms of scTM (0.73 vs. 0.69) and designability (0.85 vs. 0.80), using scTM > 0.5 as the criterion.

5.5 Ablation Studies

We conduct ablation experiments on CATH v4.2 to verify the impact of key design choices, and present the results in Table 3.

[2]ProteinMPNN, with its order-agnostic modeling, can be viewed as an autoregressive diffusion model [20].

https://doi.org/10.52202/079017-1260

5.5.1 Prior

We investigate two training strategies distinguished by their prior: 1) the structure encoder and the PLM are jointly trained; 2) the structure encoder is first pre-trained and remains frozen during the subsequent training of the PLM. We noted that the structure encoder is trained with an equivalent objective in both strategies. The latter consistently yields higher-quality protein sequences. Hence, it has been established as our default configuration.

5.5.2 Training objective

We find that the proposed simplified cross-entropy loss works better than the variational lower bound loss [24], demonstrating that the inferior performance of the vanilla Markov bridge model may stem from a harder optimization.

5.5.3 Network architecture

We observe that the performance of Bridge-IF further increases ($57.92\% \rightarrow 58.59\%$) when we replace the vanilla AdaLN with the proposed variant AdaLN-Bias. We highlight the use of AdaLN-Bias to enhance compatibility with pre-trained parameters when modulating a pre-trained Transformer model with additional conditions.

6 Conclusion

In this work, we introduce Bridge-IF, the first diffusion bridge model based on the Markov bridge process for inverse protein folding. Bridge-IF can gradually generate high-quality protein sequences from a deterministic prior. Bridge-IF achieves state-of-the-art performance in sequence recovery and foldability. **Future work** will focus on investigating more advanced structural encoders [38] and pre-training Bridge-IF using more protein structure data predicted by AlphaFold2 [28] to further enhance performance. We also intend to apply Bridge-IF to guide protein engineering aimed at designing novel functional proteins. **One potential limitation** of the proposed Bridge-IF is its lack of validation through wet-lab experiments in practical applications.

Acknowledgments and Disclosure of Funding

This research was partially supported by National Natural Science Foundation of China under grants No.12326612, Zhejiang Key R&D Program of China under grant No. 2023C03053 and No. 2024SSYS0026, Alibaba Research Intern Program.

References

[1] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pages 2023–09, 2023.

[2] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[4] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:89–113, 2021.

[5] Alexander E Chu, Tianyu Lu, and Po-Ssu Huang. Sparks of function by de novo protein design. *Nature Biotechnology*, 42(2):203–215, 2024.

[6] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[7] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

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

[9] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[10] Pat Fitzsimmons, Jim Pitman, and Marc Yor. Markovian bridges: construction, palm interpretation, and splicing. In *Seminar on Stochastic Processes, 1992*, pages 101–134. Springer, 1992.

[11] Zhangyang Gao, Cheng Tan, and Stan Z Li. Alphadesign: A graph protein design method and benchmark on alphafolddb. *arXiv preprint arXiv:2202.01079*, 2022.

[12] Zhangyang Gao, Cheng Tan, and Stan Z. Li. Pifold: Toward effective and efficient protein inverse folding. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=oMsN9TYwJ0j.

[13] Zhangyang Gao, Cheng Tan, Yijie Zhang, Xingran Chen, Lirong Wu, and Stan Z. Li. Proteininvbench: Benchmarking protein inverse folding on diverse tasks, models, and metrics. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=bqXduvuW5E.

[14] Zhangyang Gao, Cheng Tan, Xingran Chen, Yijie Zhang, Jun Xia, Siyuan Li, and Stan Z. Li. KW-design: Pushing the limit of protein design via knowledge refinement. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mpqMVWgqjn.

[15] Nate Gruver, Samuel Don Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=MfiK69Ga6p.

[16] Tymor Hamamsy, James T Morton, Robert Blackwell, Daniel Berenberg, Nicholas Carriero, Vladimir Gligorijevic, Charlie EM Strauss, Julia Koehler Leman, Kyunghyun Cho, and Richard Bonneau. Protein remote homology detection and structural alignment using deep learning. *Nature biotechnology*, pages 1–11, 2023.

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

[18] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[19] Lars Holdijk, Yuanqi Du, Priyank Jaini, Ferry Hooft, Bernd Ensing, and Max Welling. Path integral stochastic optimal control for sampling transition paths. In *ICML 2022 2nd AI for Science Workshop*, 2022.

[20] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Lm8T39vLDTE.

 https://doi.org/10.52202/079017-1260

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[22] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

[23] Po-Ssu Huang, Scott E Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, 2016.

[24] Ilia Igashov, Arne Schneuing, Marwin Segler, Michael M. Bronstein, and Bruno Correia. Retrobridge: Modeling retrosynthesis with markov bridges. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=770DetV8He.

[25] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

[26] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

[27] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1YLJDvSx6J4.

[28] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[29] Hamed Khakzad, Ilia Igashov, Arne Schneuing, Casper Goverde, Michael Bronstein, and Bruno Correia. A new age in protein design empowered by deep learning. *Cell Systems*, 14(11): 925–939, 2023.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

[32] Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems-A*, 34(4):1533–1574, 2014.

[33] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

[34] 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.

[35] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2sb: image-to-image schrödinger bridge. In *Proceedings of the 40th International Conference on Machine Learning*, pages 22042–22062, 2023.

[36] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

[37] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8): 1099–1106, 2023.

[38] Weian Mao, Muzhi Zhu, Zheng Sun, Shuaike Shen, Lin Yuanbo Wu, Hao Chen, and Chunhua Shen. De novo protein design using geometric vector field networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9UIGyJJpay.

[39] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[40] Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5 (8):1093–1109, 1997.

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[42] Milong Ren, Chungong Yu, Dongbo Bu, and Haicang Zhang. Accurate and robust protein sequence design with carbondesign. *Nature Machine Intelligence*, 6(5):536–547, 2024.

[43] Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pages 269–310, 1932.

[44] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[46] Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schrödinger bridges. In *Uncertainty in Artificial Intelligence*, pages 1985–1995. PMLR, 2023.

[47] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[48] Cheng Tan, Zhangyang Gao, Jun Xia, Bozhen Hu, and Stan Z Li. Global-context aware generative protein design. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[49] Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

 https://doi.org/10.52202/079017-1260

[53] Chuanrui Wang, Bozitao Zhong, Zuobai Zhang, Narendra Chaudhary, Sanchit Misra, and Jian Tang. Pdb-struct: A comprehensive benchmark for structure-based protein design. In *NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development*, 2023.

[54] Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. In *International conference on machine learning*, pages 10794–10804. PMLR, 2021.

[55] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[56] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[57] Kevin E Wu, Kevin K Yang, Rianne van den Berg, Sarah Alamdari, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *Nature communications*, 15(1):1059, 2024.

[58] Kai Yi, Bingxin Zhou, Yiqing Shen, Pietro Lio, and Yu Guang Wang. Graph denoising diffusion for inverse protein folding. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=u4YXKKG5dX.

[59] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, pages 40001–40039. PMLR, 2023.

[60] Mingze Yin, Hanjing Zhou, Yiheng Zhu, Miao Lin, Yixuan Wu, Jialu Wu, Hongxia Xu, Chang-Yu Hsieh, Tingjun Hou, Jintai Chen, et al. Multi-modal clip-informed protein editing. *arXiv preprint arXiv:2407.19296*, 2024.

[61] Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

[62] Lingxiao Zhao, Xueying Ding, Lijun Yu, and Leman Akoglu. Improving and unifying discrete&continuous-time discrete denoising diffusion. *arXiv preprint arXiv:2402.03701*, 2024.

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

[64] Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei Ye, and Quanquan Gu. Structure-informed language models are protein designers. In *International Conference on Machine Learning*, pages 42317–42338. PMLR, 2023.

[65] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FKksTayvGo.

[66] Yiheng Zhu, Zitai Kong, Jialu Wu, Weize Liu, Yuqiang Han, Mingze Yin, Hongxia Xu, Chang-Yu Hsieh, and Tingjun Hou. Generative ai for controllable protein sequence design: A survey. *arXiv preprint arXiv:2402.10516*, 2024.

A Algorithms

The overall workflow of the training and sampling process are provided in Algorithm 1 and Algorithm 2.

Algorithm 1 Training of the Bridge-IF

Input: coupled sample $(\boldsymbol{s}, \boldsymbol{y}) \sim p_{\mathcal{S},\mathcal{Y}}$, structure encoder $\mathcal{E}$, neural network φ_θ
$x = \mathcal{E}(s)$ ▷ Deterministic mapping from structure to sequence
$t \sim \mathcal{U}(0, \ldots, T-1),\ \boldsymbol{z}_t \sim \text{Cat}\left(\boldsymbol{z}_t; \overline{\boldsymbol{Q}}_{t-1}\boldsymbol{x}\right)$ ▷ Sample time step and intermediate state
$\hat{\boldsymbol{y}} \leftarrow \varphi_\theta(\boldsymbol{z}_t, t)$ ▷ Output of φ_θ is a vector of probabilities
$p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y}) \leftarrow \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\boldsymbol{y})\boldsymbol{z}_t\right)$ ▷ Reference transition distribution
$q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) \leftarrow \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\hat{\boldsymbol{y}})\boldsymbol{z}_t\right)$ ▷ Approximated transition distribution
Minimize $D_{\text{KL}}\left(p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{y}) \| q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)\right)$

Algorithm 2 Sampling

Input: starting point $\boldsymbol{s} \sim p_{\mathcal{S}}$, structure encoder $\mathcal{E}$, neural network φ_θ
$\boldsymbol{z}_0 \leftarrow \mathcal{E}(s)$
for t in $0, ..., T-1$:
  $\hat{\boldsymbol{y}} \leftarrow \varphi_\theta(\boldsymbol{z}_t, t)$ ▷ Output of φ_θ is a vector of probabilities
  $q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t) \leftarrow \text{Cat}\left(\boldsymbol{z}_{t+1}; \boldsymbol{Q}_t(\hat{\boldsymbol{y}})\boldsymbol{z}_t\right)$ ▷ Approximated transition distribution
  $\boldsymbol{z}_{t+1} \sim q_\theta(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)$
Return $\boldsymbol{z}_T$

B Additional results

B.1 Multi-chain protein complex design

Studying protein sequence design for multi-chain assemble structures is crucial for drug design. Next, we assess the capabilities of designing multi-chain complexes using the PDB dataset curated by Dauparas et al. [6], where sequences were clustered at 30% identity, resulting in 25,361 clusters. Following the standard data splitting, we divided those clusters randomly into three groups for training (23,358), validation (1,464), ensuring that neither the chains from the target chain nor the chains from the biounits of the target chain would be present in the other two groups.

Table 4: Performance on multi-chain protein complex dataset (in median recovery). Results of the original ProteinMPNN and GVP-Transformer were obtained using publicly available checkpoints.

Models	**Rec.** (↑)
ProteinMPNN [6]	50.00
ProteinMPNN + CMLM [ProtMPNN-CMLM]	54.39
LM-Design (ProtMPNN-CMLM + ESM-1b 650M)	59.10
LM-Design (pretrained ProtMPNN-CMLM: *freeze*)	59.43
LM-Design (pretrained ProtMPNN-CMLM: *fine-tune*)	59.43
LM-Design (ProtMPNN-CMLM + ESM-2 650M)	59.81
Bridge-IF (pretrained PiFold:*freeze* + ESM-2 650M)	61.26

As shown in Table 4, Bridge-IF also achieves similar improvements when extending to the PDB dataset, further validating its effectiveness and generalizability.

These results show that Bridge-IF can not only design single-chain proteins, which are mostly studied in previous works but also be used for designing multi-chain protein complexes.

https://doi.org/10.52202/079017-1260

C Derivations for the variational bound of reparameterized Markov bridge models

We derive the variational bound on negative log-likelihood $\log q_\theta(\boldsymbol{y}|\boldsymbol{x})$ as discussed in Section 4.2.

$$\begin{aligned}
-\log q_\theta(\boldsymbol{y}|\boldsymbol{x}) &= -\log q_\theta(\boldsymbol{z}_T|\boldsymbol{z}_0) \\
&= -\log \int q_\theta(\boldsymbol{z}_{1:T}, v_{1:T}|\boldsymbol{z}_0)\, dv_{1:T}\, d\boldsymbol{z}_{1:T-1} \\
&= -\log \int \frac{p(\boldsymbol{z}_{1:T}, v_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T)}{p(\boldsymbol{z}_{1:T}, v_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T)} q_\theta(\boldsymbol{z}_{1:T}, v_{1:T}|\boldsymbol{z}_0)\, dv_{1:T}\, d\boldsymbol{z}_{1:T-1} \\
&\leq -\int p(\boldsymbol{z}_{1:T}, v_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T) \log \frac{q_\theta(\boldsymbol{z}_{1:T}, v_{1:T}|\boldsymbol{z}_0)}{p(\boldsymbol{z}_{1:T}, v_{1:T}|\boldsymbol{z}_0, \boldsymbol{z}_T)}\, dv_{1:T}\, d\boldsymbol{z}_{1:T-1} \\
&= T \cdot \mathbb{E}_{t\sim\mathcal{U}(0,\ldots,T-1)} \mathcal{L}_t(\theta)
\end{aligned}$$

where

$$\begin{aligned}
&\mathcal{L}_t(\theta) \\
&= \mathbb{E}_{p(\boldsymbol{z}_t|\boldsymbol{x},\boldsymbol{y})} \left[\mathbb{E}_{p(\boldsymbol{v}_t)} [\mathrm{KL}(p(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t, \boldsymbol{z}_T)||q_\theta(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t))] + \mathrm{KL}\left(p(v_t)\|q_\theta(v_t)\right) \right].
\end{aligned}$$

We adopt the simplifying assumption that $q_\theta(v_t) = p(v_t)$, then $\mathcal{L}_t(\theta)$ can be written as

$$\mathcal{L}_t(\theta) = \mathbb{E}_{p(\boldsymbol{z}_t|\boldsymbol{x},\boldsymbol{y})p(v_t)} [\mathrm{KL}(p(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t, \boldsymbol{z}_T)||q_\theta(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t))], \tag{11}$$

in which the KL divergence has the form

$$\mathrm{KL}[p(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t, \boldsymbol{z}_T)||q_\theta(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t)] = \begin{cases} (1-\beta_t)\mathrm{KL}(\boldsymbol{y}||\varphi_\theta(\boldsymbol{z}_t, t)) & \text{if } v_t = 1 \\ \mathrm{KL}(\boldsymbol{z}_t||\boldsymbol{z}_t) = 0 & \text{if } v_t = 0 \end{cases} \tag{12}$$

Given that $\mathrm{KL}(\boldsymbol{y}||\varphi_\theta(\boldsymbol{z}_t, t)) = -\boldsymbol{y}^T \log \varphi_\theta(\boldsymbol{z}_t, t)$, we have

$$\begin{aligned}
&\mathbb{E}_{p(v_t)} [\mathrm{KL}\left[(p(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t, \boldsymbol{z}_T)||q_\theta(\boldsymbol{z}_{t+1}|v_t, \boldsymbol{z}_t))\right] \\
&= -(1-\beta_t) v_t \boldsymbol{y}^T \log \varphi_\theta(\boldsymbol{z}_t, t)
\end{aligned}$$

D Broader impacts

Inverse protein folding models, operating within the broader realm of bioinformatics and computational biology, have significant impacts across various scientific and practical domains. These models, by enabling the design or prediction of protein sequences that fold into specific three-dimensional structures, foster advancements in numerous fields. The broader impacts encompass several areas, including drug discovery, enzyme design, and synthetic biology.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, including the contributions made in the paper and important assumptions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-1260

Justification: See §4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See §5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code and pre-trained models will be made publicly available upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See §5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See §5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-1260

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See §5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See §5.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-1260

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.