Rationalized All-Atom Protein Design with Unified Multi-Modal Bayesian Flow

Hanlin Wu[1] **Yuxuan Song**[1,2] **Zhe Zhang**[1,2] **Zhilong Zhang**[1]
Hao Zhou[1✉] **Wei-Ying Ma**[1] **Jingjing Liu**[1]
[1] Institute of AI Industry Research (AIR), Tsinghua University
[2] Dept. of Comp. Sci. & Tech., Tsinghua University
wuhl24@mails.tsinghua.edu.cn,
{songyuxuan,zhouhao,jjliu}@air.tsinghua.edu.cn

Abstract

Designing functional proteins is a critical yet challenging problem due to the intricate interplay between backbone structures, sequences, and side-chains. Current approaches often decompose protein design into separate tasks, which can lead to accumulated errors, while recent efforts increasingly focus on all-atom protein design. However, we observe that existing all-atom generation approaches suffering from an information shortcut issue, where models inadvertently infer sequences from side-chain information, compromising their ability to accurately learn sequence distributions. To address this, we introduce a novel rationalized information flow strategy to eliminate the information shortcut. Furthermore, motivated by the advantages of Bayesian flows over differential equation–based methods, we propose the first Bayesian flow formulation for protein backbone orientations by recasting orientation modeling as an equivalent hyperspherical generation problem with antipodal symmetry. To validate, our method delivers consistently exceptional performance in both peptide and antibody design tasks. Our code, checkpoint, and designed PDBs can be found in https://github.com/GenSI-THUAIR/ProBayes.

1 Introduction

Proteins are fundamental to life, carrying out essential functions in biological processes [Clark and Pazdernik, 2012, Whitford, 2013]. Designing functional protein stands as a grand scientific challenge with great potential to modulate life processes Notin et al. [2024]. Prevalent approaches to protein design often decompose this task into a set of more manageable sub-problems, *i.e.* generating partial protein components (*e.g.* backbone), and subsequently designing the remaining components (*e.g.* sequence and side-chain) conditioned on those generated sub-units [Watson et al., 2023, Yim et al., 2023a, Bose et al., 2023, Lin and AlQuraishi, 2023].

However, in such a loosely connected pipeline, each compartment can only capture a partial protein component, which may lead to an incongruous whole. Furthermore, the three-dimensional structure of protein is simultaneously determined by both the backbone and the side chain [Schulz and Schirmer, 2013] while each model in pipelines can only capture and design a partial protein component. Moreover, early fixed inaccurate partial components accumulate and propagate errors downstream [Kong et al., 2023]. These issues all call for a unified end-to-end all-atom protein design model.

Understandably, achieving all-atom modeling is inherently challenging, due to dependency between the sequence and the coordinates (namely, the number and arrangement of side-chain atoms vary with the sequence. *e.g.* Lysine residue has four side-chain rigid groups while the Valine residue only has

[1]Correspondence to Hao Zhou (zhouhao@air.tsinghua.edu.cn).

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

one, see Fig. 1). We observe that existing all-atom modeling practices [Li et al., 2024, Chu et al., 2024, Kong et al., 2023, Zhu et al., 2024] suffer from an **information shortcut** issue: when the network is trained to jointly denoise both the sequence and the side chain, the network can infer sequence from even noised side-chain for hacking the training loss, thereby failing to properly learn the sequence distribution. In this paper, we propose to address this by designing a rationalized information flow, which avoids such problem while still preserving network's ability to capture all-atom geometry.

Furthermore, current works employ stochastic differential equations (SDE) (*e.g.* diffusion models [Song et al., 2020, Ho et al., 2020]) or ordinary differential equations (ODE) (*e.g.* flow matching [Lipman et al., 2022, Chen and Lipman, 2023]) to generate each protein component. However, these SDE and ODE methods are known to be prone to **discretization errors**, which can hinder both the quality and efficiency of the generation process [Karras et al., 2022, Tong et al., 2024, Sabour et al., 2024]. Bayesian Flow Networks (BFNs) [Graves et al., 2023] are a new class of generative models that build a generation process through Bayesian inferences over noised data, which effectively circumvents the discretization error without the need to discretize continuous differential equations [1] and has been verified to be effective in molecule generation tasks [Song et al., 2023, Wu et al., 2025].

Motivated by this, we introduce ProBayes, a new all-atom protein design model with Bayesian flows. A key theoretical challenge needs to be tackled: The orientations of the protein backbone frame are SO(3) matrices $\{\boldsymbol{R} : \boldsymbol{R} \in \mathbb{R}^{3\times3},\ \boldsymbol{R}\boldsymbol{R}^T = I,\ \det(\boldsymbol{R}) = \boldsymbol{I}\}$, while there is no existing Bayesian flow workable on SO(3). Directly conducting Bayesian inference on SO(3) is impractical, because current SO(3) distributions (*e.g.* IGSO(3)) cannot guarantee conjugacy under Bayesian inference, which is pivotal to build Bayesian flow. To overcome this issue, we propose to transform the problem into an equivalent hypersphere one, by generating SO(3) elements on hypersphere with antipodal symmetry. Additionally, ProBayes demonstrates clear advantages over the current prevalent SO(3) flow matching method [Yim et al., 2023a], which relies on a hand-crafted inference annealing technique to adjust the generation flow. Our contributions can be summarized as follows:

- We introduce ProBayes, a new protein generation approach with Bayesian flow, by transforming SO(3) generation into an equivalent problem of hypersphere generation with antipodal symmetry.
- We identify and resolve the issues of information shortcut in all-atom protein generation with a novel rationalized information flow.
- Extensive experiments demonstrate that ProBayes consistently achieves significant improvements over existing methods on peptide design and antibody design tasks, *e.g.* achieving a DockQ score of 0.74 on PepBench.

2 Preliminary

Hypersphere and hypertorus Given an Euclidean space $\mathbb{R}^d$, the hypertorus $\mathbb{T}^d$ is represented as the Cartesian product of d flat toruses $\mathbb{T}^1 = \{\vartheta : \vartheta \in [0, 2\pi)\}$. The hypersphere $\mathbb{S}^{d-1}$ located in $\mathbb{R}^d$ is defined as: $\mathbb{S}^{d-1} = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^d, ||\boldsymbol{x}||_2 = 1\}$. Superscript d denotes the degree of freedom.

Protein representation A protein $\mathcal{P}$ is a biomolecule composed of L amino acid residues, each defined by its type, backbone frame, and side-chain angles Fisher [2001]. All amino-acid types form the protein sequence $\mathcal{S} = [s_1, \ldots, s_L]$, where $s_l \in \{1, \ldots, 20\}, \forall l \in \{1, 2, \ldots, L\}$. The backbone frame of each residue is parameterized by the position vector of the C_α atom, $\boldsymbol{p} \in \mathbb{R}^3$, an orientation matrix

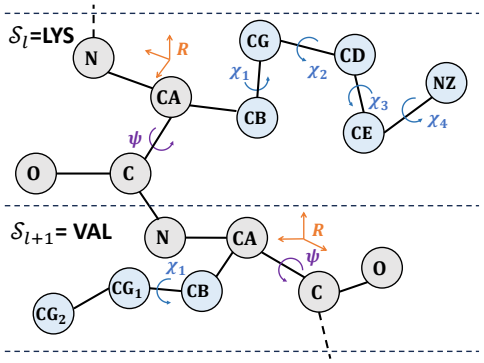


Figure 1: Illustration of protein representation. Note the pattern that different residue types have different numbers of side chain torsion angles.

[1] There exists a study [Xue et al., 2024] attempting to connect continuous-time BFNs with SDEs. However, we clarify that the discrete-time BFN functions as gradient-free generative models involving no differential operations, which is fundamentally different from SDEs. Further discussion is provided in Appendix G.

$\boldsymbol{R} \in \mathrm{SO}(3)$ that determines the relative positions of the N and C atoms, and a torsion angle ψ which governs the position of the O atom [Engh and Huber, 2006, Jumper et al., 2021, Yim et al., 2023b]. The side-chain can be represented by up to four rigid groups with $\boldsymbol{\chi}$ denoting the flexible torsion angle between rigid groups. The number of side chain torsions and arrangements depend on the specific residue type as shown in Fig. 1.

Bayesian flow networks As a new class of generative models, Bayesian flow networks (BFNs) [Graves et al., 2023] build a generation process through Bayesian inferences using noised data, which is fundamentally different from diffusion models [Song and Ermon, 2019, Ho et al., 2020, Song et al., 2020] and flow matching [Lipman et al., 2022].

Formally, BFN parameterizes a belief for clean data $\boldsymbol{x}$ over the space with a distribution family, *e.g.* Gaussian distribution for $\boldsymbol{x} \in \mathbb{R}$. The belief starts from an uninformative prior $\boldsymbol{\theta}_0$, which is updated according to the received noised data $\boldsymbol{y}_i \sim p_S(\boldsymbol{y}_i|\boldsymbol{x}, \alpha_i), i \in \{1, 2, \ldots, n\}$ with different levels of signal-to-noise ratio parameter α_i. The update transition from $\boldsymbol{\theta}_{i-1}$ to $\boldsymbol{\theta}_i$ is termed Bayesian update function $\boldsymbol{\theta}_i = h(\boldsymbol{\theta}_{i-1}, \boldsymbol{y}_i, \alpha_i)$, which is derived from the Bayesian theorem according to the form of the belief distribution and noise distribution p_S. With each $\boldsymbol{\theta}_{i-1}$ as input, the neural network Ψ is trained to transform the belief $\boldsymbol{\theta}_{i-1}$ as receiver distribution p_R, to minimize the evidence lower-bound based training loss:

$$\mathcal{L}(\Psi) = n\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\mathbb{E}_{i\sim U\{1,n\}}\mathbb{E}_{\boldsymbol{\theta}_{i-1}\sim p_F} D_{KL}[p_S(\boldsymbol{y}_i|\boldsymbol{x}, \alpha_i)||p_R(\boldsymbol{y}_i \mid \Psi(\boldsymbol{\theta}_{i-1}), \alpha_i)] \tag{1}$$

where p_F is the Bayesian flow distribution representing the distribution of $\boldsymbol{\theta}$ at time step i. Compared to diffusion models and flow matching, BFN enjoys the following advantages: (1) free from discretization error without the need to discretize continuous differential equations; (2) precisely adjusted step sizes according to the entropy parameter α_i, allowing fast and accurate generation demonstrated in [Song et al., 2023, Wu et al., 2025]. Motivated by these advantages, we aim to design a new $\mathrm{SO}(3)$ Bayesian flow as a more effective and efficient approach to protein design.

3 Bayesian flow for SO(3) generation

In this section, we introduce the first known Bayesian flow for generating SO(3) data,[2] which is derived from constructing a Bayesian Flow on the hypersphere $\mathbb{S}^3$ with antipodal symmetry.

3.1 Theoretical challenge in directly building SO(3) Bayesian flow

Difficulty in guaranteeing Bayesian conjugacy The Bayesian inference approach used in BFN requires the Bayesian conjugacy of the distribution, *i.e.* the posterior distribution after observing the noised data should be in the same distribution family as the prior belief distribution. However, existing SO(3) distributions (*e.g.* the IGSO(3) distribution used in Yim et al. [2023a,b], Bose et al. [2023]) are intractable to derive an analytical Bayesian update to ensure its conjugacy due to its infinite sum. Furthermore, other well-known existing statistical distributions over rotation matrix [Watson, 1966, Downs, 1972, Khatri and Mardia, 1977] also involve intractable probability density function to guarantee their Bayesian conjugacy, *e.g.* matrix Fisher distribution.

Therefore, we resort to other widely used rotation representations such as Euler angles, axis-angle and unit quaternions, instead of the rotation matrix. As discussed below, we demonstrate that both Euler angles and the axis-angle representation fail to adequately capture the group structure of SO(3), whereas unit quaternions are able to do so effectively, which is used in this paper. Additionally, the unit quaternion's intrinsic manifold, *i.e.* the hypersphere, is significantly simpler than the structure of SO(3), and there exists a normal distribution over the hypersphere which satisfies Bayesian conjugacy.

Non-covering of Euler angle and axis-angle representation We observe that some rotation representations cannot capture the group structure of SO(3). For example, Euler angles suffer from the singularity problem of gimbal lock [Hoag, 1963]. From the perspective of topology, such problems reveal the fact that the hyper-torus $\mathbb{T}^3$ cannot *cover* the SO(3). Practically, we validate the ineffectiveness of Euler angles and axis-angle representation via toy data generation experiment (see Appendix C). The definition of the cover map is as follows:

[2] Although the position vector and the orientation matrix form $\mathrm{SE}(3)$, measure on $\mathrm{SE}(3)$ can be disintegrated into measure on $\mathrm{SO}(3)$ and $\mathbb{R}^3$ [Yim et al., 2023a, Bose et al., 2023]. Therefore, we can focus on $\mathrm{SO}(3)$ generation only.

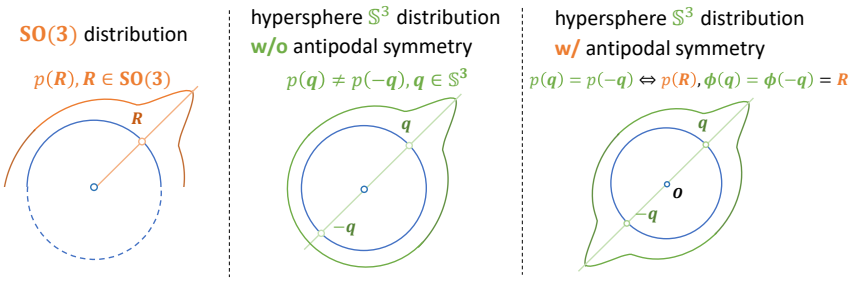


Figure 2: Intuitive illustration of the equivalent transformation from SO(3) generation to hypersphere generation with antipodal symmetry. Extending the modeled space to a hypersphere yields a more symmetric and tractable manifold.

Definition 1 (Covering Map, Hatcher [2005])**.** *Given topological space X and Y, a covering map is a surjective continuous map $f : X \to Y$ such that: every element $y \in Y$ has an open neighborhood $U_y \subseteq Y$ where $f^{-1}(U_y)$ is a disjoint union of open sets, each of which is mapped by f homeomorphically onto U_y.*

The quaternion representation of rotation can be used to build an SO(3) Bayesian flow because the space of this representation (*i.e.* the hypersphere) can *cover* SO(3). Intuitively, we aim to build a generative model in a larger topological space while preserving the original group structure of SO(3).

3.2 Transforming SO(3) generation into hypersphere $\mathbb{S}^3$ generation

Fig. 2 provides an intuitive illustration of transforming SO(3) generation into hypersphere $\mathbb{S}^3$ generation. Firstly, we introduce the group structure of unit quaternion rotation representation, *i.e.* SU(2) group and its corresponding manifold hypersphere $\mathbb{S}^3$. Then, we explain the proposed key constraint of antipodal symmetry to *equivalently* transform SO(3) generation into hypersphere $\mathbb{S}^3$ generation, and introduce a simple approach to satisfy such constraint.

SU(2) group SU(2) is the special unitary group composed by unitary matrices $\boldsymbol{U} \in \{\boldsymbol{U} : \boldsymbol{U} \in \mathbb{C}^{2\times 2},\ \boldsymbol{U}\boldsymbol{U}^\dagger = I,\ \det(\boldsymbol{U}) = 1\}$. SU(2) is diffeomorphic to the hypersphere $\mathbb{S}^3$ [Hall, 2013]. Hence, we can represent SU(2) elements using four real numbers, namely the quaternion $\boldsymbol{q} = [a, b, c, d]^T$ with unit 2-norm constraint: $\mathrm{SU}(2) = \{\begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix} : a, b, c, d \in \mathbb{R}, a^2+b^2+c^2+d^2 = 1\}$

Double covering between SO(3) and SU(2) In fact, SU(2) is the double covering group of SO(3) [Hall, 2013] and there exists a surjective homomorphism $\phi : \mathrm{SU}(2) \to \mathrm{SO}(3)$ such that for every $g_{\mathrm{SU}(2)} \in \mathrm{SU}(2)$, we have $\phi(g_{\mathrm{SU}(2)}) = \phi(-g_{\mathrm{SU}(2)}) = g_{\mathrm{SO}(3)} \in \mathrm{SO}(3)$. Therefore, we can transform the representations of these two groups between one another. Specifically, given a unit quaternion $\boldsymbol{q} = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$, the corresponding rotation matrix $\boldsymbol{R} = \phi(\boldsymbol{q}) = \phi(-\boldsymbol{q})$ can be obtained by:

$$\boldsymbol{R} = \begin{pmatrix} 1-2c^2-2d^2 & 2bc-2da & 2bd+2ac \\ 2bc+2da & 1-2b^2-2d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & 1-2b^2-2c^2 \end{pmatrix} \text{ [Shoemake, 1985]} \tag{2}$$

Inversely, rotation matrix $\boldsymbol{R}$ can be transformed into the corresponding double unit quaternion $\boldsymbol{q}_1 = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$ and $\boldsymbol{q}_2 = -(a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k})$ from Eq. (2):

$$s = \frac{1}{2\sqrt{1+\mathrm{tr}(\boldsymbol{R})}}, a = \frac{1}{4s}, b = s(\boldsymbol{R}_{32} - \boldsymbol{R}_{23}), c = s(\boldsymbol{R}_{13} - \boldsymbol{R}_{31}), d = s(\boldsymbol{R}_{21} - \boldsymbol{R}_{12}) \tag{3}$$

As illustrated above, each rotation matrix $\boldsymbol{R}$ corresponds to two unit quaternions $\boldsymbol{q}$ and $-\boldsymbol{q}$, which are equivalent when representing the three-dimensional rotation operation. This phenomenon is termed as SU(2)'s double-covering of SO(3) [Altmann, 2005], which results in the *discrepancy* between generating rotation matrices and directly generating unit quaternions. We propose a natural solution to this problem, by adding an antipodal symmetry constraint to the modeled unit quaternion distribution on the hypersphere $\mathbb{S}^3$:

Definition 2 (Antipodal Symmetry)**.** *For a distribution $p(\boldsymbol{x})$ supported on $\mathbb{S}^d$, we say this distribution is antipodal invariant if $p(\boldsymbol{x}) = p(-\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{S}^d$. Correspondingly, we say a function f operating $\boldsymbol{q} \in \mathbb{S}^3$ is antipodal equivariant if $f(-\boldsymbol{q}) = -f(\boldsymbol{q})$.*

Now, with antipodal symmetry, we can transform the generative modeling problem of SO(3) data into an equivalent one on $\mathbb{S}^3$ with antipodal symmetry. We formulate the proposed transformation with the following preposition:

Proposition 1. *Every distribution $p_{SO(3)}(\boldsymbol{R})$ supported on SO(3) can be bijectively mapped to a distribution $p_{\mathbb{S}^3}(\boldsymbol{q})$ supported on $\mathbb{S}^3$ which satisfies $p_{\mathbb{S}^3}(\boldsymbol{q}) = p_{\mathbb{S}^3}(-\boldsymbol{q})$.*

In fact, we find that antipodal equivariance can be easily ensured by implementing the Invariant Point Attention (IPA) module [Jumper et al., 2021] based on quaternion representation (instead of the rotation matrix) without extra operations or complexity.

3.3 Bayesian flow on hypersphere $\mathbb{S}^d$

Equipped with Proposition 3, we can now derive a Bayesian flow for generating SO(3) on the hypersphere $\mathbb{S}^3$. In this section, we introduce the first derivation of the Bayesian flow on $\mathbb{S}^d$, which can be applied to generate unit quaternions with $d = 3$ for SO(3) generation.

Von Mises Fisher distribution and Bayesian update function h We choose von Mises Fisher (vMF) distribution [Mardia and Jupp, 2009] to parameterize the belief over $\mathbb{S}^d$ because of its tractability and Bayesian conjugacy. With the location parameter μ and concentration parameter κ, the probability density function of the von Mises–Fisher distribution is:

$$f_p(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = vMF(\boldsymbol{x}|\boldsymbol{\mu}, \kappa) = C_p(\kappa) \exp(\kappa \boldsymbol{\mu}^T \boldsymbol{x}) = \frac{\kappa^\nu}{(2\pi)^{\nu+1} I_\nu(\kappa)} \exp(\kappa \boldsymbol{\mu}^T \boldsymbol{x}), \tag{4}$$

where $||\boldsymbol{\mu}|| = ||\boldsymbol{x}|| = 1, \nu = d/2 - 1, \kappa \geq 0$, $I_\nu(\kappa)$ denotes the modified Bessel function of the first kind at order ν. Given the prior belief parameterized by vMF distribution with parameter $\boldsymbol{\mu}_{i-1}$, κ_{i-1}, the noised sample $\boldsymbol{y}_i$ from sender distribution with unknown clean $\boldsymbol{x}$, and the accuracy α_i describing the entropy of $\boldsymbol{y}_i$, the Bayesian update function h for von Mises Fisher distribution is deducted as:

$$h(\{\boldsymbol{\mu}_{i-1}, \kappa_{i-1}\}, \boldsymbol{y}_i, \alpha_i) = \{\boldsymbol{\mu}_i, \kappa_i\}, \text{where } \kappa_i \boldsymbol{\mu}_i = \kappa_{i-1} \boldsymbol{\mu}_{i-1} + \alpha_i \boldsymbol{y}_i \tag{5}$$

The proof of the above Eq. (5) can be found in Appendix B. This update function operates in an intuitive manner, analogous to the moment equation in physics, where the accuracy/entropy parameters κ and α represent the mass, and $\boldsymbol{\mu}$ and $\boldsymbol{y}$ denote the velocity. We define $\boldsymbol{\theta} \overset{\text{def}}{=} \kappa \boldsymbol{\mu}$ to compactly denote the parameter of the vMF distribution $\{\kappa, \boldsymbol{\mu}\}$.

Bayesian flow distribution Based on the Bayesian update function, the Bayesian flow distribution on hypersphere $\mathbb{S}^d$ over vMF distribution can be obtained by conducting a series of Bayesian inferences. With the notation $\alpha_{1:i} \overset{\text{def}}{=} \{\alpha_1, \alpha_2, \ldots, \alpha_i\}$, the Bayesian flow distribution is expressed as (details in Appendix B):

$$p_F(\boldsymbol{\theta}_i|\boldsymbol{x}; \alpha_{1:i}) = \mathop{\mathbb{E}}_{vMF(\boldsymbol{y}_1|\boldsymbol{x},\alpha_1)\ldots vMF(\boldsymbol{y}_i|\boldsymbol{x},\alpha_i)} \delta(\boldsymbol{\theta}_i - \sum_{j=1}^{i} \alpha_j \boldsymbol{y}_j) \tag{6}$$

Now with Eq. (6) and Proposition 3, we are able to generate the orientation of the backbone frame with a Bayesian flow, which is free from discretization error and enjoys entropy-adjusted dynamic step size. The antipodal symmetry of the modeled distribution is guaranteed as follows:

Proposition 2. *With Ψ as an antipodal equivariant function,* i.e. *$\Psi(-\boldsymbol{\theta}^{\boldsymbol{q}}) = -\Psi(\boldsymbol{\theta}^{\boldsymbol{q}})$, and uniform prior $p(\boldsymbol{\theta}_0^{\boldsymbol{q}})$ on $\mathbb{S}^3$, the marginal distribution defined by multiple Markov transitions $p_\Psi(\boldsymbol{\theta}_n^{\boldsymbol{q}}) = p(\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int \Pi_{i=1}^n p_F(\boldsymbol{\theta}_i^{\boldsymbol{q}}|\Psi(\boldsymbol{\theta}_{i-1}^{\boldsymbol{q}}); \alpha_{1:i}) d\boldsymbol{\theta}_{1:n-1}^{\boldsymbol{q}}$ is antipodal invariant, which is equivalent to a distribution $p_\Psi(\boldsymbol{R})$.*

Note that, as the rotation and orientation are ubiquitous in geometric deep learning, such SO(3) Bayesian flow can be used in other applications, *e.g.*, molecular docking [Corso et al., 2022].

4 Rationalized all-atom protein design

In this section, we present our approach to jointly generating the complete protein structure, including the sequence, backbone and sidechain. We emphasize that such a co-design process is inherently tricky, and naive joint training risks information shortcut between sequence and side-chain.

ProBayes parameterizes the belief over the protein space via each modality's belief parameter $\boldsymbol{\theta}^{\mathcal{P}} = \{\boldsymbol{\theta}^{\mathcal{S}}, \boldsymbol{\theta}^{\boldsymbol{p}}, \boldsymbol{\theta}^{\boldsymbol{R}}, \boldsymbol{\theta}^{\psi}, \boldsymbol{\theta}^{\boldsymbol{\chi}}\}$ with a predefined number of generation steps n. For training, we first sample the time index $i \sim U\{1, \dots, n\}$ and $\boldsymbol{\theta}_{i-1} \sim p_F$ for each modality. The neural network $\Psi(\boldsymbol{\theta}^{\mathcal{P}}_{i-1}, i)$ is trained to minimize the training loss according to Eq. (1). After sufficient training, the data distribution p_{data} is well captured by the neural network Ψ. As for sampling, each modality starts from an uninformative prior $\boldsymbol{\theta}^{\mathcal{P}}_0$, which is to be updated according to the network output $\Psi(\boldsymbol{\theta}^{\mathcal{P}}, i)$ and the Bayesian flow distribution till $i = n$.

4.1 Bayesian flow for each protein modality

We explain the definition of the Bayesian flow for each protein modality. For detailed training and sampling algorithm, please see Appendix D.

Bayesian flow for orientation matrix $\boldsymbol{R}$ generation We utilize the proposed Bayesian flow Eq. (6) to generate the orientation of the protein backbone frame $\boldsymbol{R}$. We begin by uniformly converting the orientation matrix $\boldsymbol{R}$ into its corresponding quaternion representation, yielding one of the two equivalent quaternions $\{\boldsymbol{q}, -\boldsymbol{q}\}$ with equal probability. The SO(3) Bayesian flow distribution is:

$$p_F^{\boldsymbol{q}}(\boldsymbol{\theta}_i^{\boldsymbol{q}} | \boldsymbol{R}; \alpha_{1:i}^{\boldsymbol{q}}) = \mathop{\mathbb{E}}_{\boldsymbol{q} \sim U\{\phi^{-1}(\boldsymbol{R})\}} \mathop{\mathbb{E}}_{vMF(\boldsymbol{y}_1 | \boldsymbol{q}, \alpha_1^{\boldsymbol{q}}) \dots vMF(\boldsymbol{y}_i | \boldsymbol{q}, \alpha_i^{\boldsymbol{q}})} \delta(\boldsymbol{\theta}_i^{\boldsymbol{q}} - \sum_{j=1}^{i} \alpha_j^{\boldsymbol{q}} \boldsymbol{y}_j) \tag{7}$$

where $\alpha_i^{\boldsymbol{q}}$ is the predefined accuracy to make the entropy of Eq. (7) linearly decrease with respect to time step i. Sampling $\boldsymbol{\theta}_i^{\boldsymbol{q}}$ from $p_F^{\boldsymbol{q}}$, the training objective is derived based on Eq. (1) and the KL divergence between vMF distributions (details in Appendix B):

$$\mathcal{L}(\Psi) = n \mathop{\mathbb{E}}_{\boldsymbol{q} \sim p_{\text{data}}, i \sim U\{1,n\}, \boldsymbol{\theta}_i^{\boldsymbol{q}} \sim p_F^{\boldsymbol{q}}} \alpha_i^{\boldsymbol{q}} \frac{I_{\nu+1}(\alpha_i^{\boldsymbol{q}})}{I_\nu(\alpha_i^{\boldsymbol{q}})} (1 - \text{dot}(\boldsymbol{q}, \Psi^{\boldsymbol{q}}(\boldsymbol{\theta}^{\mathcal{P}}, i))) \tag{8}$$

where dot represents the dot product between two quaternions, and $\Psi^{\boldsymbol{q}}$ represents the quaternion part of the network prediction. We prove in Appendix B that such a loss is proportional to the geodesic distance between the predicted and ground truth orientation matrix up to a constant, which further solidifies the transformation between SO(3) and SU(2).

Bayesian flow for C_α position $\boldsymbol{p}$ generation The C_α positions of the backbone frame are in Euclidean space $\boldsymbol{p} \in \mathbb{R}^3$. Therefore, we use the Gaussian distribution Bayesian flow [Graves et al., 2023] to generate them. The Bayesian flow distribution of C_α position $\boldsymbol{p}$ is:

$$p_F^{\boldsymbol{p}}(\boldsymbol{\mu}^{\boldsymbol{p}} | \boldsymbol{p}; t) = \mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{p}} | \gamma(t)\boldsymbol{p}, \gamma(t)(1 - \gamma(t))\boldsymbol{I}), \text{where } \gamma(t) = 1 - \sigma_1^{2t} \tag{9}$$

where σ_1 is the hyperparameter controlling the entropy of the Bayesian flow at $t = 1$. The variance of the belief Gaussian distribution $\ln \sigma^2(t) = \ln(1 - \gamma(t))/t$ does not need to be modeled because it is deterministic *w.r.t.* time steps.

Bayesian flow for sequence $\mathcal{S}$ generation Given that sequences are discrete variables, we utilize the discrete Bayesian flow [Graves et al., 2023] to generate the sequence. With the notation of projection from the class index j to the length K one-hot vector $(\mathbf{e}_j)_k \overset{\text{def}}{=} \delta_{jk}$, where $\mathbf{e}_j \in \mathbb{R}^K, \mathbf{e}_{\mathcal{S}} \overset{\text{def}}{=} (\mathbf{e}_{s_1}, \dots, \mathbf{e}_{s_L}) \in \mathbb{R}^{K \times L}$, the Bayesian flow distribution of sequence $\mathcal{S}$ is defined as:

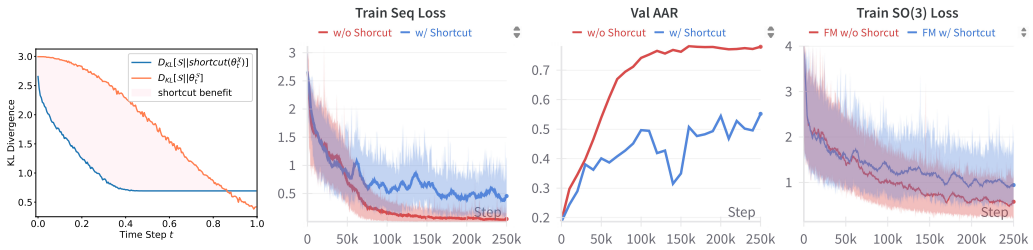
$$p_F^{\mathcal{S}}(\boldsymbol{\theta}^{\mathcal{S}} \mid \mathcal{S}; t) = \mathop{\mathbb{E}}_{\mathcal{N}(\boldsymbol{y} | \beta^{\mathcal{S}}(t)(K\mathbf{e}_{\mathcal{S}} - \mathbf{1}_{K \times L}), \beta^{\mathcal{S}}(t) K \boldsymbol{I}_{K \times L \times L})} \delta\left(\boldsymbol{\theta}^{\mathcal{S}} - \frac{e^{\boldsymbol{y}} \boldsymbol{\theta}_0^{\mathcal{S}}}{\sum_{k=1}^{K} e^{\boldsymbol{y}_k} (\boldsymbol{\theta}_0)_k^{\mathcal{S}}}\right) \tag{10}$$

where $\beta^{\mathcal{S}}(t)$ is the predefined accuracy schedule.

Bayesian flow for angles $\psi, \boldsymbol{\chi}$ generation The torsional angles ψ and $\boldsymbol{\chi}$ are periodic variables on the hypertorus. Hence, we use the periodic Bayesian flow [Wu et al., 2025] to generate them:

$$p_F(\boldsymbol{m}_i^{\boldsymbol{\chi}} | \boldsymbol{\chi}; \alpha_{1:i}^{\boldsymbol{\chi}}) = \mathop{\mathbb{E}}_{vM(\boldsymbol{y}_1 | \boldsymbol{\chi}, \alpha_1^{\boldsymbol{\chi}}) \dots vM(\boldsymbol{y}_i | \boldsymbol{\chi}, \alpha_i^{\boldsymbol{\chi}})} \delta(\boldsymbol{m}_i^{\boldsymbol{\chi}} - \text{atan2}(\sum_{j=1}^{i} \alpha_j^{\boldsymbol{\chi}} \cos \boldsymbol{y}_j, \sum_{j=1}^{i} \alpha_j^{\boldsymbol{\chi}} \sin \boldsymbol{y}_j)) \tag{11}$$

We illustrate our method using $\boldsymbol{\chi}$; the Bayesian flow for ψ has the same form. While both lie on the same manifold, unlike Li et al. [2024], we treat them differently. Specifically, we distinguish $\boldsymbol{\chi}$ from the backbone torsion ψ because $\boldsymbol{\chi}$ carries partial sequence information. We expand on this in the next section.



(a) (b) (c) (d)

Figure 3: (a): Illustration of information shortcut measured by KL divergence. (b)-(d): Training curves demonstrating the negative impact of the shortcut. As shown in Fig. 3b, the network with a shortcut achieves a smaller loss initially, but exhibits a larger loss towards the end. As for the metric, the shortcut results in a worse validation AAR in Fig. 3c. Additionally, we observe that the shortcut also hurts the performance of other modalities, such as the SO(3) in Fig. 3d.

4.2 The information shortcut problem

As discussed in Sec. 2 and Fig. 1, the residue type $\mathcal{S}$ reflects the types of the side-chain with varying numbers of side-chain torsion angles from 0 to 4 [Jumper et al., 2021, Li et al., 2024]. However, diffusion-like models (*e.g.* FM and BFNs) are inflexible in capturing varying numbers of random variables. Prior practices [Li et al., 2024, Zhu et al., 2024] tried to address this by padding the number of variables with a fixed padding value (*e.g.* 0 in [Li et al., 2024]), then mixing the side chain information with the sequence and backbone.

Shortcut induced by paddings However, we prove that such padding and mixing operations cause information leakage from the *even noised* side chain information to the sequence prediction. To illustrate the idea, we design a simple threshold-based shortcut function to "predict" the ground truth sequence for residue Alanine and Glycine, which have no side chain torsion angles *i.e.* $\boldsymbol{\chi} = [0, 0, 0, 0]$:

$$\texttt{shortcut}(\boldsymbol{\theta}_t^{\boldsymbol{\chi}}) = \begin{cases} [\frac{1}{2}, \frac{1}{2}, 0, \dots, 0] & \text{if } \texttt{circmean}(\boldsymbol{\theta}^{\boldsymbol{\chi}}) < 0.5 \\ [\frac{1}{20}, \frac{1}{20}, \frac{1}{20}, \dots, \frac{1}{20}] & \text{otherwise} \end{cases} \tag{12}$$

where `circmean` is the circular mean of periodic variables [Mardia and Jupp, 2009], with the first two elements representing the predicted probability of Alanine and Glycine, respectively. We compare the KL divergence between the predicted categorical distribution $\texttt{shortcut}(\theta_t^{\boldsymbol{\chi}})$ and the ground-truth one-hot distribution $\mathcal{S}$, against the KL divergence between the noised sequence θ_t^s and $\mathcal{S}$, as shown in Fig. 3a. The results demonstrate that the shortcut prediction achieves significantly lower KL divergence than the noised sequence most of the time. This indicates the network can exploit shortcut information from the number of side-chain angles/paddings to minimize sequence training loss, rather than genuinely denoising the sequence by learning the underlying sequence distribution.

Shortcut induced by inherent side-chain distributions Moreover, we emphasize that the information shortcut can still exist even in the absence of padding. This is because the side-chain torsion distributions themselves (i.e., the non-padded values) are inherently dependent on the residue types, as shown in Fig. 4. Consequently, the network can exploit these statistical regularities in the side-chain distributions to reduce the sequence prediction loss, thereby hindering the proper learning of the true sequence distribution.

Rationalized information flow Therefore, we propose a rationalized network information flow to avoid the possibility of shortcuts, as illustrated in Fig. 5. Concretely, we avoid involving side-chain information with the sequence $\Psi^{\mathcal{S}} = \Psi(\boldsymbol{\theta}^{\mathcal{S}}, \boldsymbol{\theta}^{\boldsymbol{p}}, \boldsymbol{\theta}^{\boldsymbol{R}}, \boldsymbol{\theta}^{\psi}, \cancel{\boldsymbol{\theta}^{\boldsymbol{\chi}}})$, while maintaining network's capacity to capture the all-atom geometry by the backbone and side-chain mixing module for denoising other modalities. For side-chain prediction, the loss is computed only for non-padded values to prevent the network from learning harmful patterns. The detailed implementation is provided in Appendix F.

Excluding side-chain information from sequence prediction does not negatively impact the performance ceiling of sequence generation. This stems from the fundamental unfair relationship between sequence and side-chain in the data distribution: The sequence serves as the primary determinant of residue types, while side-chain configurations are dependent on the sequence for both side-chain atom types and conformational specifications.

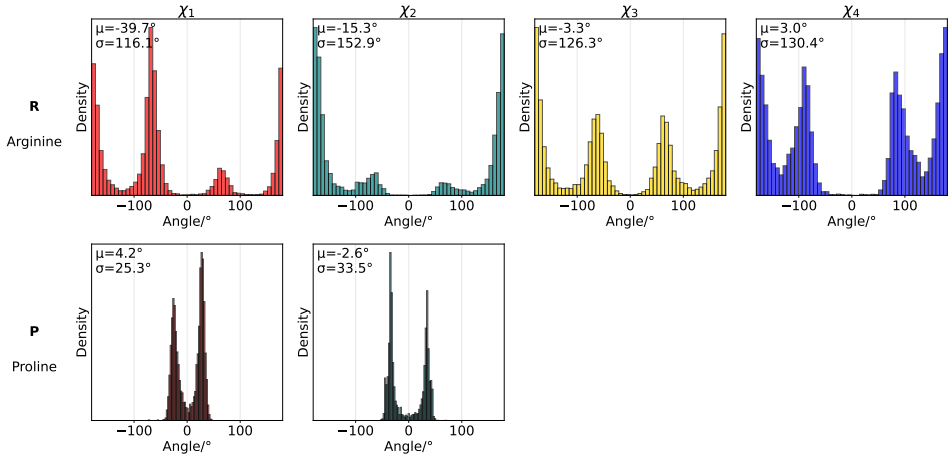


Figure 4: Visualization of the side-chain angle distributions by residue type in PepBench. The side-chain angle distributions of the two residues differ significantly, and the network can similarly exploit the statistical characteristics of side-chain distributions to "predict" the sequence just like Eq. (12).

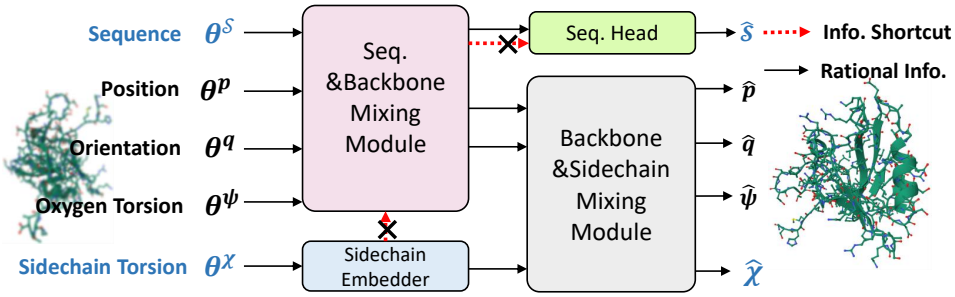


Figure 5: Illustration of the proposed rationalized information flow.

5 Related work

Protein generation There is a body of meaningful protein generation practices focusing on single protein component, *e.g.* backbone only Yim et al. [2023b], Bose et al. [2023], Geffner et al. [2025], Watson et al. [2023], sequence only Madani et al. [2020], Dauparas et al. [2022], and side-chain only Zhang et al. [2023]. Recently, motivated by the advantages of joint modeling, an increasing number of sequence–structure co-design approaches have emerged, *e.g.*, backbone–sequence co-design [Campbell et al., 2024, Lisanza et al., 2024] and all-atom protein generation [Martinkus et al., 2024, Li et al., 2024, Chu et al., 2024, Kong et al., 2024, Zhu et al., 2024]. However, the increased information of all-atom geometry also brings challenges: We observe that all-atom generation is prone to the information shortcut problem due to the dependency between sequence and side-chain, which is highlighted and resolved in this paper.

BFN for molecule generation Inspired by BFN's unified generative modeling for continuous and discrete data, GeoBFN [Song et al., 2023] first incorporates molecular geometric constraints into the BFN framework and achieves exceptional generation performance and efficiency. Recently, Wu et al. [2025] proposed to build BFN on the hypertorus for periodic material generation, demonstrating a superior sampling efficiency. However, the rigidity of the protein backbone requires generative modeling on SO(3), which is inherently difficult due to the intractability of ensuring Bayesian conjugacy with SO(3) distributions. In this paper, we address the problem by transforming the modeled space into a simpler one, thereby allowing Bayesian inferences.

Table 1: Evaluation on peptide co-design. On each target, 40 candidates are generated for evaluation.

Dataset	Method	Energy		Native Likeness		Design		
		$\Delta G(\downarrow)$	Success(↑)	DockQ(↑)	$\text{RMSD}_{C_\alpha}(\downarrow)$	Valid (↑)	V&Div(↑)	V&Novel(↑)
PepBench	Test Set	-35.25	95.70	1.0	0.0	0.989	-	-
	RFDiffusion	-24.61	69.86	0.265	4.12	0.943	0.439	0.670
	PepFlow	-23.09	59.20	0.677	2.81	0.848	**0.453**	0.623
	PepGLAD	-21.94	55.97	0.656	3.35	0.330	0.245	0.226
	ProBayes	**-28.77**	**72.85**	**0.742**	**2.27**	**0.998**	0.449	**0.727**
PepBDB	Test Set	-35.96	95.79	1.0	0.0	-	-	-
	PepFlow	-22.45	65.60	0.517	4.71	0.615	0.379	0.521
	PepGLAD	-24.53	48.47	0.472	5.44	0.165	0.163	0.128
	ProBayes	**-30.55**	**85.78**	**0.605**	**3.39**	**0.959**	**0.423**	**0.725**

Table 2: Evaluation on binding conformation generation. Baseline results are from Kong et al. [2024].

Model	PepBench			PepBDB		
	$\text{RMSD}_{C_\alpha}(\downarrow)$	$\text{RMSD}_{\text{atom}}(\downarrow)$	DockQ(↑)	$\text{RMSD}_{C_\alpha}(\downarrow)$	$\text{RMSD}_{\text{atom}}(\downarrow)$	DockQ(↑)
AlphaFold 2	8.49	9.20	0.355	-	-	-
DiffAb	4.23	7.60	0.586	13.96	13.12	0.236
PepGLAD	4.09	5.30	0.592	8.87	8.62	0.403
ProBayes	**2.62**	**3.81**	**0.672**	**3.64**	**4.91**	**0.563**

6 Experiments

We verify the effectiveness of ProBayes on two protein design tasks, including peptide design and antibody design. For peptide design task, our evaluation configurations follow Kong et al. [2024], including peptide co-design and peptide binding conformation generation. The description of metrics and implementation details can be found in Appendix F. We also include a toy dataset experiment in Appendix C of SO(3) Bayesian flow compared to SDE / ODE methods [Bose et al., 2023].

6.1 Peptide design

Datasets The datasets used for evaluation include **PepBench** and **PepBDB** [Wen et al., 2019]. **PepBench** is constructed from the Protein Data Bank [Berman et al., 2000], containing 6105 complexes and the LNR dataset Tsaban et al. [2022] is utilized as the test set with 93 complexes. **PepBDB** is a protein-peptide complex dataset containing 8434, 370, and 190 items for training, validation, and test, respectively.

Baselines We evaluate our approach against the following baselines: **PepFlow** [Li et al., 2024], a peptide design model utilizing Riemannian flow matching; **PepGLAD** [Kong et al., 2024], which employs latent diffusion techniques for all-atom peptide modeling; and **RFDiffusion**, a multi-stage protein design framework [Watson et al., 2023] integrated with ProteinMPNN [Dauparas et al., 2022] for sequence generation.

Results We report the results on PepBench and PepBDB in Table 1 and Table 2 for seq-structure co-design and binding conformation generation task, respectively. Across both PepBench and PepBDB datasets, ProBayes outperforms existing methods in energy, native structural fidelity, and design quality. For peptide binding conformation generation task, our method significantly outperforms all baselines. It achieves the lowest RMSD_{C_α}, $\text{RMSD}_{\text{atom}}$, and DockQ, indicating ProBayes's superior all-atom structure generation accuracy.

6.2 Antibody design

In this experiment, we focus on designing the third Complementarity-Determining Region of the antibody heavy chain (CDR-H3), as it exhibits the greatest variability and plays a dominant role in antigen binding [MacCallum et al., 1996].

Dataset We utilize the SAbDab database [Dunbar et al., 2014] of antibody–antigen complexes as our training dataset and evaluate model performance using the RAbD bench-

Table 3: Results of antibody design task on RAbD.

Method	AAR ↑	RMSD ↓	E_{total} ↓	ΔG ↓
Test Set	100.0	0.00	-16.76	-15.33
dyMEAN	40.05	2.36	1239.29	612.75
DiffAb	35.04	2.53	495.69	489.42
AbX	41.27	2.40	281.29	237.22
AbDPO	31.29	2.79	270.12	116.06
AbDPO++	36.25	2.48	338.14	223.73
ProBayes	**42.12**	**2.27**	**54.68**	**44.12**

mark [Adolf-Bryfogle et al., 2018]. The data-processing procedures, evaluation configurations, and baseline results are from [Ye et al., 2024]

Baselines We compare our method against three existing approaches: **dyMEAN** [Kong et al., 2023] employs a full-atom geometric encoder with iterative non-autoregressive generation; **DiffAb** [Luo et al., 2022] jointly diffuses categorical residue types, C_α coordinates, and residue orientations; **AbDPO** and its variant **AbDPO++** [Zhou et al., 2024] are methods tailored for optimizing antibody energy based on direct preference optimization.

Results The results are listed in Tab. 3. With consistently better AAR and RMSD, ProBayes also demonstrates significant improvements in energy-based metrics, achieving state-of-the-art results with a lower order of magnitude, surpassing even the energy-optimization-focused models AbDPO and AbDPO++.

6.3 Ablation study

Table 4: Results of ablation study on the antibody design task.

Method	AAR↑	RMSD↓	E_{total} ↓	ΔG ↓
ProBayes	**42.12%**	**2.27**	**54.68**	**44.12**
w/o anti. symm.	40.17%	2.40	171.33	104.15
w/ shortcut	32.51%	2.42	203.29	86.76
w/ Rosetta pack	42.12%	2.27	76.11	49.28

We validate our design considerations through the following ablation experiments using the antibody design task. Results can be found in Tab. 4: (1) **antipodal symmetry**: we break the model's antipodal symmetry by removing equivariance in the Markov transition *i.e.* we replace our quaternion-based implementation with the original IPA implementation [Jumper et al., 2021], which leads to a significant drop in energy-related metrics, demonstrating the necessity of enforcing the proposed constraints. (2)**information shortcut**: we reintroduce the information shortcut by incorporating side-chain information during sequence prediction. This experiment results in a ∼10% drop in AAR and hurts performance across other metrics, indicating the detrimental impact of such shortcuts on model performance. (3) **all-atom capability**: we replace model-generated side-chains with those obtained from Rosetta's side-chain packing [Alford et al., 2017]. Our model achieves superior performance on energy-based metrics, demonstrating its capability to generate high-fidelity side-chain structures.

7 Conclusion

We introduce ProBayes, a novel all-atom protein design model using Bayesian flows, which effectively resolves critical issues of information shortcut that plagues existing methods. Achieving significant improvements over prior methods on peptide and antibody design tasks, ProBayes demonstrates a more robust pathway towards generating functional proteins.

Acknowledgments

This work is supported by the National Science and Technology Major Project (2022ZD0117502), the National Natural Science Foundation of China (Grant No. 62376133), the Wuxi Research Institute of Applied Technologies, Tsinghua University under Grant 20242001120.

References

David P Clark and Nanette J Pazdernik. *Molecular biology*. Elsevier, 2012.

David Whitford. *Proteins: structure and function*. John Wiley & Sons, 2013.

Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nature biotechnology*, 42(2):216–228, 2024.

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.

Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023a.

Avishek Joey Bose, Tara Akhound-Sadegh, Kilian Fatras, Guillaume Huguet, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *arXiv preprint arXiv:2301.12485*, 2023.

Georg E Schulz and R Heiner Schirmer. *Principles of protein structure*. Springer Science & Business Media, 2013.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. End-to-end full-atom antibody design. *arXiv preprint arXiv:2302.00203*, 2023.

Jiahan Li, Chaoran Cheng, Zuofan Wu, Ruihan Guo, Shitong Luo, Zhizhou Ren, Jian Peng, and Jianzhu Ma. Full-atom peptide design based on multi-modal flow matching. *arXiv preprint arXiv:2406.00735*, 2024.

Alexander E Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024.

Tian Zhu, Milong Ren, and Haicang Zhang. Antibody design using a score-based diffusion model guided by evolutionary, physical and geometric constraints. In *Forty-first International Conference on Machine Learning*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ricky TQ Chen and Yaron Lipman. Riemannian flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Vinh Tong, Trung-Dung Hoang, Anji Liu, Guy Van den Broeck, and Mathias Niepert. Learning to discretize denoising diffusion odes. *arXiv preprint arXiv:2405.15506*, 2024.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. *arXiv preprint arXiv:2404.14507*, 2024.

Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks. *arXiv preprint arXiv:2308.07037*, 2023.

Kaiwen Xue, Yuhao Zhou, Shen Nie, Xu Min, Xiaolu Zhang, Jun Zhou, and Chongxuan Li. Unifying bayesian flow networks and diffusion models through stochastic differential equations. *arXiv preprint arXiv:2404.15766*, 2024.

Yuxuan Song, Jingjing Gong, Hao Zhou, Mingyue Zheng, Jingjing Liu, and Wei-Ying Ma. Unified generative modeling of 3d molecules with bayesian flow networks. In *The Twelfth International Conference on Learning Representations*, 2023.

Hanlin Wu, Yuxuan Song, Jingjing Gong, Ziyao Cao, Yawen Ouyang, Jianbing Zhang, Hao Zhou, Wei-Ying Ma, and Jingjing Liu. A periodic bayesian flow for material generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Lz0XW99tE0.

Matthew Fisher. Lehninger principles of biochemistry, ; by david l. nelson and michael m. cox. *The Chemical Educator*, 6:69–70, 2001.

RA Engh and R Huber. Structure quality and target parameters. 2006.

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.

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. *arXiv preprint arXiv:2302.02277*, 2023b.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Geoffrey S Watson. The statistics of orientation data. *The Journal of Geology*, 74(5, Part 2):786–797, 1966.

Thomas D Downs. Orientation statistics. *Biometrika*, 59(3):665–676, 1972.

CG Khatri and Kanti V Mardia. The von mises–fisher matrix distribution in orientation statistics. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 39(1):95–106, 1977.

David Hoag. Apollo guidance and navigation: Considerations of apollo imu gimbal lock. *Canbridge: MIT Instrumentation Laboratory*, pages 1–64, 1963.

Allen Hatcher. *Algebraic topology*. Tsinghua University Press, 2005.

Brian C Hall. Lie groups, lie algebras, and representations. In *Quantum Theory for Mathematicians*, pages 333–366. Springer, 2013.

Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

Simon L Altmann. *Rotations, quaternions, and double groups*. Courier Corporation, 2005.

Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. *arXiv preprint arXiv:2503.00710*, 2025.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

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.

Yangtian Zhang, Zuobai Zhang, Bozitao Zhong, Sanchit Misra, and Jian Tang. Diffpack: A torsional diffusion model for autoregressive protein side-chain packing. *arXiv preprint arXiv:2306.01794*, 2023.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel WK Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J Hendel, Miriam K Simma, Ge Liu, Muna Yase, Hongwei Wu, et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, pages 1–11, 2024.

Karolis Martinkus, Jan Ludwiczak, Kyunghyun Cho, Wei-Ching Liang, Julien Lafrance-Vanasse, Isidro Hotzel, Arvind Rajpal, Yan Wu, Richard Bonneau, Vladimir Gligorijevic, and Andreas Loukas. Abdiffuser: Full-atom generation of in vitro functioning antibodies, 2024. URL https://arxiv.org/abs/2308.05027.

Xiangzhe Kong, Yinjun Jia, Wenbing Huang, and Yang Liu. Full-atom peptide design with geometric latent diffusion. *arXiv preprint arXiv:2402.13555*, 2024.

Zeyu Wen, Jiahua He, Huanyu Tao, and Sheng-You Huang. Pepbdb: a comprehensive structural database of biological peptide–protein interactions. *Bioinformatics*, 35(1):175–177, 2019.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

Tomer Tsaban, Julia K Varga, Orly Avraham, Ziv Ben-Aharon, Alisa Khramushin, and Ora Schueler-Furman. Harnessing protein folding neural networks for peptide–protein docking. *Nature communications*, 13(1):176, 2022.

Robert M MacCallum, Andrew CR Martin, and Janet M Thornton. Antibody-antigen interactions: contact analysis and binding site topography. *Journal of molecular biology*, 262(5):732–745, 1996.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42 (D1):D1140–D1146, 2014.

Jared Adolf-Bryfogle, Oleks Kalyuzhniy, Michael Kubitz, Brian D Weitzner, Xiaozhen Hu, Yumiko Adachi, William R Schief, and Roland L Dunbrack Jr. Rosettaantibodydesign (rabd): A general framework for computational antibody design. *PLoS computational biology*, 14(4):e1006112, 2018.

Fei Ye, Zaixiang Zheng, Dongyu Xue, Yuning Shen, Lihao Wang, Yiming Ma, Yan Wang, Xinyou Wang, Xiangxin Zhou, and Quanquan Gu. Proteinbench: A holistic evaluation of protein foundation models. *arXiv preprint arXiv:2409.06744*, 2024.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

Xiangxin Zhou, Dongyu Xue, Ruizhe Chen, Zaixiang Zheng, Liang Wang, and Quanquan Gu. Antigen-specific antibody design via direct energy-based preference optimization. *Advances in Neural Information Processing Systems*, 37:120861–120891, 2024.

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.

Daniel Frisch and Uwe D Hanebeck. Deterministic von mises–fisher sampling on the sphere using fibonacci lattices. In *2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*, pages 1–8. IEEE, 2023.

James A Brofos, Marcus A Brubaker, and Roy R Lederman. Manifold density estimation via generalized dequantization. *arXiv preprint arXiv:2102.07143*, 2021.

Michiel Hazewinkel, Nadiya Gubareni, and Vladimir V Kirichenko. *Algebras, Rings and Modules: Volume 1*, volume 575. Springer Science & Business Media, 2006.

Sankar Basu and Björn Wallner. Dockq: a quality measure for protein-protein docking models. *PloS one*, 11(8):e0161879, 2016.

Haitao Lin, Odin Zhang, Huifeng Zhao, Dejun Jiang, Lirong Wu, Zicheng Liu, Yufei Huang, and Stan Z Li. Ppflow: Target-aware peptide design with torsional flow matching. *bioRxiv*, pages 2024–03, 2024.

Contents

A Proofs of propositions

Proposition 3. *Every distribution $p_{SO(3)}(\boldsymbol{R})$ supported on SO(3) can be bijectively mapped to a distribution $p_{\mathbb{S}^3}(\boldsymbol{q})$ supported on $\mathbb{S}^3$ which satisfies $p_{\mathbb{S}^3}(\boldsymbol{q}) = p_{\mathbb{S}^3}(-\boldsymbol{q})$.*

Proof. Given each distribution $p_{SO(3)}(\boldsymbol{R})$ support on the SO(3) group, we can build the corresponding $\mathbb{S}^3$ measure $p_{SO(3)}(\boldsymbol{R})$ using the surjective homomorphism $\phi : \mathrm{SU}(2) \to \mathrm{SO}(3)$:

$$\forall \boldsymbol{q} \in \mathbb{S}^3, p_{\mathbb{S}^3}(\boldsymbol{q}) := p_{SO(3)}(\phi^{-1}(\boldsymbol{R}))/2 \tag{13}$$

Now we prove that $p_{\mathbb{S}^3}(\boldsymbol{q})$ satisfies all the probability distribution axioms and the antipodal symmetry:

1. Non-negativity: $\forall \boldsymbol{q} \in \mathbb{S}^3, p_{\mathbb{S}^3}(\boldsymbol{q}) = p_{SO(3)}(\phi^{-1}(\boldsymbol{R}))/2 \geq 0$
2. Normalization: $\int_{\mathbb{S}^3} p_{\mathbb{S}^3}(\boldsymbol{q}) d\mathbb{S}^3 = \frac{1}{2}\int_{\mathbb{S}^3} p(\boldsymbol{R}) d\mathbb{S}^3 = \frac{1}{2}\int_{\mathbb{S}^3} p(\boldsymbol{R}) d\mathrm{SO}(3) = 1$
3. Antipodal symmetry: $p_{\mathbb{S}^3}(\boldsymbol{q}) = \frac{1}{2} p_{SO(3)}(\phi^{-1}(\boldsymbol{R})) = p_{\mathbb{S}^3}(-\boldsymbol{q})$

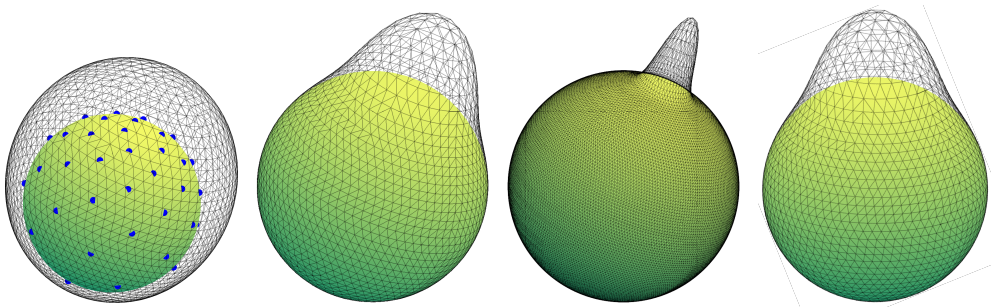
For $p_{\mathbb{S}^3}(\boldsymbol{q}) = p_{\mathbb{S}^3}(-\boldsymbol{q})$, we can also build $p_{SO(3)}(\boldsymbol{R}) := 2p_{\mathbb{S}^3}(\boldsymbol{q}) = 2p_{\mathbb{S}^3}(-\boldsymbol{q}), \forall \boldsymbol{R} \in \mathrm{SO}(3)$. We can check that $p_{SO(3)}(\boldsymbol{R}) \geq 0$ and $\int_{SO(3)} p_{SO(3)}(\boldsymbol{R}) dSO(3) = 1$. Therefore, $p_{SO(3)}$ is a probability density function on SO(3). □

Proposition 4. *With Ψ as an antipodal equivariant function,* i.e. *$\Psi(-\boldsymbol{\theta^q}) = -\Psi(\boldsymbol{\theta^q})$, and uniform prior $p(\boldsymbol{\theta}_0^{\boldsymbol{q}})$ on $\mathbb{S}^3$, the marginal distribution defined by multiple Markov transitions $p_\Psi(\boldsymbol{\theta}_n^{\boldsymbol{q}}) = p(\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int \Pi_{i=1}^n p_F(\boldsymbol{\theta}_i^{\boldsymbol{q}} | \Psi(\boldsymbol{\theta}_{i-1}^{\boldsymbol{q}}); \alpha_{1:i}) d\boldsymbol{\theta}_{1:n-1}^{\boldsymbol{q}}$ is antipodal invariant, which is equivalent to a distribution $p_\Psi(\boldsymbol{R})$.*

Proof.

$$\begin{aligned}
p_\Psi(-\boldsymbol{\theta}_n^{\boldsymbol{q}}) &= p(-\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int p_\Psi(-\boldsymbol{\theta}_{n:1}^{\boldsymbol{q}} | -\boldsymbol{\theta}_0^{\boldsymbol{q}}) d\boldsymbol{\theta}_{1:n-1}^{\boldsymbol{q}} \\
&= p(-\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int \prod_{t=0}^{n-1} p_\Psi(-\boldsymbol{\theta}_{t+1}^{\boldsymbol{q}} | -\boldsymbol{\theta}_t^{\boldsymbol{q}}) d\boldsymbol{\theta}_{1:n-1}^{\boldsymbol{q}} \\
&= p(-\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int \prod_{t=0}^{n-1} p_F(-\boldsymbol{\theta}_{t+1}^{\boldsymbol{q}} | \Psi(-\boldsymbol{\theta}_{i-1}^{\boldsymbol{q}}); \alpha_{1:i}) d\boldsymbol{\theta}_{1:n-1}^{\boldsymbol{q}} \\
&= p(-\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int \prod_{t=0}^{n-1} p_F(-\boldsymbol{\theta}_{t+1}^{\boldsymbol{q}} | -\Psi(\boldsymbol{\theta}_{i-1}^{\boldsymbol{q}}); \alpha_{1:i}) d\boldsymbol{\theta}_{1:n-1}^{\boldsymbol{q}} \\
&= p(-\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int \prod_{t=0}^{n-1} p_F(\boldsymbol{\theta}_i^{\boldsymbol{q}} | \Psi(\boldsymbol{\theta}_{i-1}^{\boldsymbol{q}}); \alpha_{1:i}) d\boldsymbol{\theta}_{1:n}^{\boldsymbol{q}} \\
&= p(\boldsymbol{\theta}_0^{\boldsymbol{q}}) \int \prod_{t=0}^{n-1} p_F(\boldsymbol{\theta}_i^{\boldsymbol{q}} | \Psi(\boldsymbol{\theta}_{i-1}^{\boldsymbol{q}}); \alpha_{1:i}) d\boldsymbol{\theta}_{1:n}^{\boldsymbol{q}} \\
&= p_\Psi(\boldsymbol{\theta}_n^{\boldsymbol{q}}).
\end{aligned}$$

Therefore, the marginal distribution $p(\boldsymbol{\theta}_n^{\boldsymbol{q}})$ is antipodal invariant. □



(a) $\kappa = 1$, $\boldsymbol{\mu} = [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]^T$ (b) $\kappa = 10$, $\boldsymbol{\mu} = [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]^T$ (c) $\kappa = 100$, $\boldsymbol{\mu} = [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]^T$ (d) $\kappa = 10$, $\boldsymbol{\mu} = [0, 0, 1]^T$

Figure 6: Visualization of the von Mises Fisher distribution with different parameters with $d = 2$. Those figures are from Frisch and Hanebeck [2023].

B Detailed derivation of SO(3) Bayesian flow

B.1 Hypersphere and von Mises Fisher distribution

Given an Euclidean space $\mathbb{R}^d$, the hypersphere $\mathbb{S}^{d-1}$ located in $\mathbb{R}^d$ is defined as:

$$\mathbb{S}^{d-1} = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^d, ||\boldsymbol{x}||_2 = 1\} \tag{14}$$

The von Mises Fisher distribution [Mardia and Jupp, 2009] is a uni-modal distribution on $\mathbb{S}^{d-1}$ with the location parameter $\boldsymbol{\mu}$ and concentration parameter κ. The parameters $\boldsymbol{\mu}$ and κ are analogous to the mean μ and variance $1/\sigma^2$ in the normal distribution: $\boldsymbol{\mu}$ represents the central location around which the distribution is concentrated, and κ serves as a measure of concentration. We provide a visualization of vMF distribution with different $\boldsymbol{\mu}$ and κ in Fig. 6. The probability density function of vMF distribution is:

$$f_p(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = vMF(\boldsymbol{x}|\boldsymbol{\mu}, \kappa) = C_p(\kappa) \exp(\kappa \boldsymbol{\mu}^T \boldsymbol{x}) = \frac{\kappa^\nu}{(2\pi)^{\nu+1} I_\nu(\kappa)} \exp(\kappa \boldsymbol{\mu}^T \boldsymbol{x}), \tag{15}$$

where $||\boldsymbol{\mu}|| = ||\boldsymbol{x}|| = 1, \nu = d/2 - 1, \kappa \geq 0$, $I_\nu(\kappa)$ denotes the modified Bessel function of the first kind at order ν. The differential entropy of $vMF(\boldsymbol{\mu}, \kappa)$ is:

$$H(vMF(\boldsymbol{\mu}, \kappa)) = -\kappa \frac{I_{\nu+1}(\kappa)}{I_\nu(\kappa)} + (\nu + 1) \ln 2\pi + \kappa + \ln I_\nu(\kappa) - \nu \ln \kappa \tag{16}$$

We use vMF's entropy Eq. (16) to determine the accuracy schedule in the following sections.

B.2 Bayesian update function h

Sender distribution For each time step i, the sender corrupts the clean data $\boldsymbol{x} \sim p_{\text{data}}$ using the sender distribution p_S and the signal-to-noise ratio parameter α_i, gets the noised information $\boldsymbol{y} \sim p_S$, and send $\boldsymbol{y}$ to the receiver. Here, we use the vMF distribution to instantiate p_S for hypersphere data $\boldsymbol{x} \in \mathbb{S}^{d-1}$:

$$p_S(\boldsymbol{y}|\boldsymbol{x}; \alpha) = vMF(\boldsymbol{y}|\boldsymbol{x}, \alpha) \tag{17}$$

Input distribution As for the receiver, its belief over the ground truth data $\boldsymbol{x}$ is formulated by a distribution family with parameter $\boldsymbol{\theta}$, *e.g.*, Gaussian distribution with μ and σ for Euclidean space data $x \in \mathbb{R}^1$, which can be expressed as:

$$p_I(\boldsymbol{x}|\boldsymbol{\theta}) = vMF(\boldsymbol{x}|\boldsymbol{\mu}, \kappa) \tag{18}$$

Bayesian update function For each time step i, the receiver updates its prior belief $\boldsymbol{\theta}_{i-1}$ according to the observed noised data $\boldsymbol{y}_i$ with known accuracy α_i according to the Bayesian theorem. This process is formulated as the Bayesian update function $\boldsymbol{\theta}_i = h(\boldsymbol{\theta}_{i-1}, \boldsymbol{y}_i, \alpha_i)$. The Bayesian update

function h for von Mises Fisher distribution is derived as follows:

$$\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{y}) &= p(\boldsymbol{y}|\boldsymbol{x};\alpha)p(\boldsymbol{x};\boldsymbol{\mu}_{i-1},\kappa_{i-1})/p(\boldsymbol{y}) \\
&\propto p(\boldsymbol{y}|\boldsymbol{x};\alpha)p(\boldsymbol{x};\boldsymbol{\mu}_{i-1},\kappa_{i-1}) \\
&= vMF(\boldsymbol{y}|\boldsymbol{x},\alpha)vMF(\boldsymbol{x}|\boldsymbol{\mu}_{i-1},\kappa_{i-1}) \\
&\propto \exp\{\alpha\boldsymbol{x}^T\boldsymbol{y}+\kappa_{i-1}\boldsymbol{\mu}_{i-1}\} \\
&= vMF(\boldsymbol{x}|\boldsymbol{\mu}_i,\kappa_i)
\end{aligned}$$

where $\kappa_i\boldsymbol{\mu}_i = \kappa_{i-1}\boldsymbol{\mu}_{i-1}+\alpha_i\boldsymbol{y}_i$. Therefore, the Bayesian update function is:

$$h(\{\boldsymbol{\mu}_{i-1},\kappa_{i-1}\},\boldsymbol{y}_i,\alpha_i) = \{\boldsymbol{\mu}_i,\kappa_i\}, \text{where } \kappa_i\boldsymbol{\mu}_i = \kappa_{i-1}\boldsymbol{\mu}_{i-1}+\alpha_i\boldsymbol{y}_i \tag{19}$$

We define $\boldsymbol{\theta} \overset{\text{def}}{=} \kappa\boldsymbol{\mu}$ to compactly denote the parameter of the vMF distribution $\{\kappa,\boldsymbol{\mu}\}$.

B.3 Bayesian flow distribution $p_F(\boldsymbol{\theta}_i|\boldsymbol{x};\alpha_{1:i})$

Given time index i and clean data $\boldsymbol{x}$, the Bayesian flow distribution for $\boldsymbol{\theta}_i$ is defined as the marginal distribution after conducting multiple Bayesian updates over noised data $\boldsymbol{y}_i$:

$$p_F(\boldsymbol{\theta}_i|\boldsymbol{x};\alpha_{1:i}) = \mathop{\mathbb{E}}_{vMF(\boldsymbol{y}_1|\boldsymbol{x},\alpha_1)\dots vMF(\boldsymbol{y}_i|\boldsymbol{x},\alpha_i)} \delta(\boldsymbol{\theta}_i - \sum_{j=1}^{i}\alpha_j\boldsymbol{y}_j) \tag{20}$$

For determining the accuracy schedule $\alpha_{1:i}$ according to entropy, we first formalize the entropy of the Bayesian flow:

$$H(t) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{p_F(\boldsymbol{\theta}_i|\,\mathbf{x};\alpha_{1:i})} H(p_I(\cdot|\boldsymbol{\theta}_i)), \text{where } i = nt+1 \tag{21}$$

The entropy of the Bayesian flow is dominated by the accuracy schedule $\alpha_{1:n}$. Although Eq. (21) is not tractable, we can evaluate its value at each step given the accuracy schedule $\alpha_{1:n}$. For synchronizing the entropy across each modality, we find the accuracy schedule that $\alpha_{1:n}$ such that the entropy of this Bayesian flow linearly decreases with respect to time:

$$H(t) = (1-t)H(0)+tH(1) \tag{22}$$

Although Eq. (22) is not analytically tractable, we can solve this equation using binary search to find each α_i at each step i.

B.4 Discrete-time loss with n steps $L^n(\mathbf{x})$

Receiver distribution Given $\boldsymbol{\theta}_i$ from the Bayesian flow distribution as input, the receiver uses the network Ψ to improve its belief considering interdependency across modalities and dimensions, which is referred to as the output distribution. We choose to parameterize the output distribution as delta distribution to directly predict the ground truth $\boldsymbol{x}$ following [Graves et al., 2023]:

$$p_O(\boldsymbol{x}|\Psi(\boldsymbol{\theta}_i,t_i)) = \delta(\boldsymbol{x}-\Psi^{\boldsymbol{x}}(\boldsymbol{\theta}_i,t_i)) \tag{23}$$

To approximate the sender's distribution, the receiver's distribution is obtained by convolving the output distribution with the sender's distribution:

$$p_R(\boldsymbol{y}|\boldsymbol{\theta}_i;t_i,\alpha_i) = \mathbb{E}_{p_O(\boldsymbol{x}'|\Psi(\boldsymbol{\theta}_i,t_i))}p_S(\boldsymbol{y}|\boldsymbol{x}';\alpha_i) = vMF(\boldsymbol{y}|\Psi^{\boldsymbol{x}}(\boldsymbol{\theta}_i,t_i),\alpha_i) \tag{24}$$

Discrete-time loss with n steps The discrete time loss is defined as the KL divergence between sender and receiver distribution:

$$L^n(\mathbf{x}) = n \mathop{\mathbb{E}}_{i\sim U\{1,n\},p_F(\boldsymbol{\theta}_i|\boldsymbol{x};\alpha_{1:i})} D_{KL}\left(p_S\left(\cdot \mid \boldsymbol{x};\alpha_i\right) \parallel p_R(\cdot \mid \boldsymbol{\theta}_i;t_i,\alpha_i)\right) \tag{25}$$

$$= n \mathop{\mathbb{E}}_{i\sim U\{1,n\},p_F(\boldsymbol{\theta}_i|\boldsymbol{x};\alpha_{1:i})} \alpha_i\frac{I_{\nu+1}(\alpha_i)}{I_\nu(\alpha_i)}(1-\text{dot}(\boldsymbol{x},\Psi(\boldsymbol{\theta},t_i))) \tag{26}$$

In fact, the term $(1-\text{dot}(\boldsymbol{x},\Psi(\boldsymbol{\theta},t_i)))$ in Eq. (25) is proportional to the geodesic distance between the corresponding orientation matrix $\boldsymbol{R} = \phi(\boldsymbol{x})$ and $\hat{\boldsymbol{R}} = \phi = (\hat{\boldsymbol{x}}) = \phi(\Psi(\boldsymbol{\theta},t_i))$.

Proposition 5. *The training loss defined in Eq. (8) is proportional to the geodesic distance between the predicted orientation matrix $\hat{\boldsymbol{R}} = \phi(\Psi^{\boldsymbol{q}}(\boldsymbol{\theta}^{\mathcal{P}}, t))$ and the ground truth orientation matrix $\boldsymbol{R}$ up to a constant.*

Proof. We first introduce the $(w, \boldsymbol{v})$ representation for unit quaternions. Specifically, for a unit quaternion $\boldsymbol{q} = [a, b, c, d]^T$, we define the scalar part as $w = a$ and the vector part as $\boldsymbol{v} = b\boldsymbol{i}+c\boldsymbol{j}+d\boldsymbol{k}$, so that the quaternion can be written as $\boldsymbol{q} = w + \boldsymbol{v}$, where $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are the standard basis elements corresponding to the three Cartesian axes. This representation is particularly convenient because the rotation angle ϑ of the rotation matrix $\boldsymbol{R} = \phi^{-1}(\boldsymbol{q})$ satisfies the relation

$$w = a = \cos(\vartheta/2) \quad \text{[Shoemake, 1985]} \tag{27}$$

Using the $(w, \boldsymbol{v})$ representation, the geodesic distance between two unit quaternions $\boldsymbol{q} = w_1 + \boldsymbol{v}_1$ and $\hat{\boldsymbol{q}} = w_2 + \boldsymbol{v}_2$ on $\mathbb{S}^3$ is given by:

$$d_{\mathbb{S}^3}(\boldsymbol{q}, \hat{\boldsymbol{q}}) = \arccos\left(\langle \boldsymbol{q}, \hat{\boldsymbol{q}} \rangle\right) = \arccos\left(w_1 w_2 + \langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle\right), \tag{28}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product.

The geodesic distance between the corresponding rotation matrices $\boldsymbol{R} = \phi(\boldsymbol{q})$ and $\hat{\boldsymbol{R}} = \phi(\hat{\boldsymbol{q}})$ in the rotation group $\mathrm{SO}(3)$ is the rotation angle of the relative rotation $\boldsymbol{R}_r = \boldsymbol{R}^\top \hat{\boldsymbol{R}}$. This angle can be computed via the quaternion representation of $\boldsymbol{R}_r$:

$$\boldsymbol{q}_r = \phi^{-1}(\boldsymbol{R}_r) = \boldsymbol{q}^{-1} \times \hat{\boldsymbol{q}} = \left[w_1 w_2 + \langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle,\ w_1 \boldsymbol{v}_2 + w_2 \boldsymbol{v}_1 + \boldsymbol{v}_1 \times \boldsymbol{v}_2\right]. \tag{29}$$

The corresponding rotation angle ϑ of $\boldsymbol{R}_r$ is then:

$$d_{SO(3)}(\boldsymbol{R}, \hat{\boldsymbol{R}}) = \vartheta = 2 \arccos\left(w_1 w_2 + \langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle\right) = 2\, d_{\mathbb{S}^3}(\boldsymbol{q}, \hat{\boldsymbol{q}}). \tag{30}$$

The proof is done. □

C SO(3) toy data experiment

In this section, we use the synthetic dataset in Brofos et al. [2021] to verify the effectiveness of the proposed SO(3) Bayesian flow over SDE/ODE based methods following [Bose et al., 2023].

Toy dataset network The antipodal equivariance of the toy dataset network is achieved by the following formula:

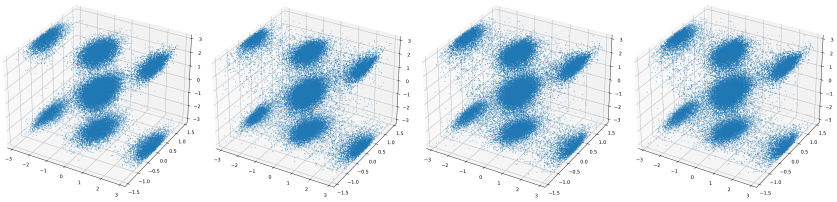
$$\Psi^{\boldsymbol{q}}_{\text{toy}}(\boldsymbol{\theta}^{\boldsymbol{q}}, t) = \text{Normalize}(\boldsymbol{\mu} \otimes \text{MLP}(\boldsymbol{\mu}\boldsymbol{\mu}^T, \kappa, t) + \boldsymbol{\mu}) \tag{31}$$

where $\otimes$ denotes the element-wise product between two vectors, and $\text{Normalize}(\boldsymbol{q}) = \boldsymbol{q}/||\boldsymbol{q}||_2$.

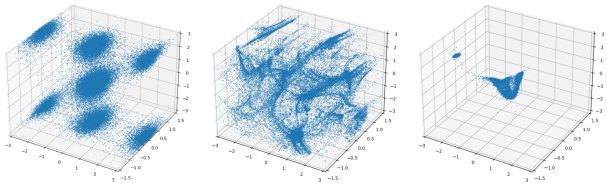
Metrics The metrics W1 and W2 refer to the Wasserstein p-distance between the generated distribution and the ground truth test distribution with $p = 1$ and $p = 2$, respectively. These two metrics measure the distributional distance between the generated distribution and the ground truth data distribution.

Results We present the visualized generated results in Fig. 7 and the corresponding evaluation metrics in Tab. 5. As shown in Fig. 7, ProBayes produces higher-quality samples with fewer outliers compared to the SDE and ODE baselines. This observation is consistent with the improved W1 and W2 scores reported in Tab. 5, further demonstrating the effectiveness of the proposed SO(3) Bayesian flow.

We further validate the ineffectiveness of the Euler angle and axis-angle rotation representations in Fig. 7f and Fig. 7g, as discussed in the main text. Specifically, we employ the hypertorus BFN [Wu et al., 2025] to generate the Euler angle representation, and a combination of the hypertorus and hypersphere BFN to produce the axis-angle representation. As shown, both representations fail to capture the underlying SO(3) group structure, resulting in poor SO(3) generative modeling performances.



(a) data dist. (b) FoldFlow-base (c) FoldFlow-OT (d) FoldFlow-SFM



(e) ProBayes (f) Euler angle (g) Axis-angle

Figure 7: The data is visualized using the Euler-angle representation of the rotation matrices. (a)-(d) are from Bose et al. [2023]

Table 5: Toy dataset performance comparison. Baseline results are from [Bose et al., 2023].

Method	W1 ($\times 10^{-2}, \downarrow$)	W2 ($\times 10^{-1}, \downarrow$)
FoldFlow-base	5.39 ± 0.88	1.52 ± 0.27
FoldFlow-OT	4.96 ± 0.27	1.25 ± 0.12
FoldFlow-SFM	4.92 ± 1.56	1.26 ± 0.49
Simulated SDE	5.13 ± 1.36	1.33 ± 0.44
ProBayes	$\mathbf{3.23} \pm 0.68$	$\mathbf{0.99} \pm 0.33$

D Detailed training and sampling procedure

We describe the training loss functions used for the remaining modalities:

$\mathbf{C}_\alpha$ position $\boldsymbol{p}$ training loss Sampling $\boldsymbol{\mu}_{i-1}^{\boldsymbol{p}}$ from $p_F^{\boldsymbol{p}}$, the training loss for position $\boldsymbol{p}$ is calculated as the discrete time loss for continuous Euclidean space variable:

$$\mathcal{L}_{\boldsymbol{p}} = \frac{n}{2}\left(1-\sigma_1^{2/n}\right) \mathop{\mathbb{E}}_{i\sim U\{1,n\}, p_F^{\boldsymbol{p}}(\boldsymbol{\mu}_{i-1}^{\boldsymbol{p}}|\boldsymbol{p};t_{i-1})} \frac{\left\|\boldsymbol{p}-\Psi_{i-1}^{\boldsymbol{p}}(\boldsymbol{\theta}^{\mathcal{P}},t)\right\|^2}{\sigma_1^{2i/n}} \tag{32}$$

Sequence $\mathcal{S}$ training loss Sampling $\boldsymbol{\theta}_i^{\mathcal{S}}$ from $p_F^{\mathcal{S}}$, the training loss is the n-step discrete-time loss for discrete variable:

$$\begin{aligned}\mathcal{L}_{\mathcal{S}} = n \mathop{\mathbb{E}}_{i\sim U\{1,n\}, p_F^{\mathcal{S}}(\boldsymbol{\theta}^{\mathcal{S}}|\mathcal{S};t_{i-1}), \mathcal{N}(\mathbf{y}|\alpha_i(K\mathbf{e}_{\mathcal{S}}-\mathbf{1}),\alpha_i K\boldsymbol{I})} & \ln\mathcal{N}\left(\mathbf{y} \mid \alpha_i\left(K\mathbf{e}_{\mathcal{S}}-\mathbf{1}\right), \alpha_i K\boldsymbol{I}\right) \\ & - \sum_{d=1}^{L} \ln\left(\sum_{k=1}^{K} p_O^{(d)}(k \mid \boldsymbol{\theta}^{\mathcal{S}};t_{i-1}) \mathcal{N}\left(y^{(d)} \mid \alpha_i\left(K\mathbf{e}_k-\mathbf{1}\right), \alpha_i K\boldsymbol{I}\right)\right)\end{aligned} \tag{33}$$

where $\boldsymbol{I} \in \mathbb{R}^{K\times L\times L}$ and $\mathbf{1} \in \mathbb{R}^{K\times D}$.

Angle $\boldsymbol{\chi}, \psi$ training loss Sampling $\boldsymbol{\theta}^{\boldsymbol{\chi}}$ from $p_F^{\boldsymbol{\chi}}$, the training loss is the n-step discrete-time loss for periodic variable:

$$\mathcal{L}_{\boldsymbol{\chi}} = n \mathop{\mathbb{E}}_{i\sim U\{1,n\}, p_F(\boldsymbol{\theta}^{\boldsymbol{\chi}}|\boldsymbol{\chi};\alpha_{1:i})} \alpha_i \frac{I_1(\alpha_i)}{I_0(\alpha_i)}(1-\cos(\boldsymbol{\chi}-\Psi(\boldsymbol{\theta}_{i-1}^{\mathcal{P}},t_{i-1})) \tag{34}$$

Note that for $\boldsymbol{\chi}$, we only calculate the loss for those non-padded values. Also, sampling $\boldsymbol{\theta}^{\psi}$ from p_F^{ψ}, the training loss for ψ is:

$$\mathcal{L}_{\psi} = n \mathop{\mathbb{E}}_{i\sim U\{1,n\}, p_F(\boldsymbol{\theta}^{\boldsymbol{\chi}}|\psi;\alpha_{1:i})} \alpha_i \frac{I_1(\alpha_i)}{I_0(\alpha_i)}(1-\cos(\psi-\Psi(\boldsymbol{\theta}_{i-1}^{\mathcal{P}},t_{i-1})) \tag{35}$$

Algorithm 1 Training

Require: $n, K \in \mathbb{Z}, \sigma_1 \in \mathbb{R}^+, \beta_1 \in \mathbb{R}^+, \alpha^{\boldsymbol{\chi}}_{1:n}, \alpha^{\psi}_{1:n}, \alpha^{\boldsymbol{q}}_{1:n} \in \mathbb{R}^+, \lambda_{\boldsymbol{q}}, \lambda_{\mathcal{S}}, \lambda_{\boldsymbol{p}}, \lambda_{\boldsymbol{\chi}}, \lambda_{\psi}, \lambda_{bb} \in \mathbb{R}^+$
Input: sequence $\mathcal{S}$, position $\boldsymbol{p}$, orientation $\boldsymbol{R}$, backbone torsion ψ, sidechain torsion $\boldsymbol{\chi}$, $\boldsymbol{\chi}$ mask $\text{Mask}^{\boldsymbol{\chi}}$
Sample $i \sim U\{1, n\}$, $t \leftarrow \frac{(i-1)}{n}$
randomly select one from the two quaternions
$\boldsymbol{q} \sim U\{\phi^{-1}(\boldsymbol{R})\}$
sampling from the Bayesian flow distribution of each modality
$\boldsymbol{\theta}^{\boldsymbol{q}} \sim p^{\boldsymbol{q}}_F(\boldsymbol{\theta}^{\boldsymbol{q}}_i | \boldsymbol{R}; \alpha^{\boldsymbol{q}}_{1:i})$
$\boldsymbol{\theta}^{\mathcal{S}} \sim p^{\mathcal{S}}_F(\boldsymbol{\theta}^{\mathcal{S}} \mid \mathcal{S}; t)$
$\boldsymbol{\mu}^{\boldsymbol{p}} \sim p^{\boldsymbol{p}}_F(\boldsymbol{\mu}^{\boldsymbol{p}} | \boldsymbol{p}; t), \boldsymbol{\theta}^{\boldsymbol{p}} \leftarrow \{\boldsymbol{\mu}^{\boldsymbol{p}}\}$
$\boldsymbol{m}^{\boldsymbol{\chi}}_i \sim p^{\boldsymbol{\chi}}_F(\boldsymbol{m}^{\boldsymbol{\chi}}_i | \boldsymbol{\chi}; \alpha^{\boldsymbol{\chi}}_{1:i}), \boldsymbol{c}^{\boldsymbol{\chi}}_i \sim p^{\boldsymbol{\chi}}_F(\boldsymbol{c}^{\boldsymbol{\chi}}_i | \boldsymbol{\chi}; \alpha^{\boldsymbol{\chi}}_{1:i})$,
$\boldsymbol{m}^{\boldsymbol{\chi}}_i \leftarrow \boldsymbol{m}^{\boldsymbol{\chi}}_i \cdot \text{Mask}^{\boldsymbol{\chi}}(\Psi^{\mathcal{S}}), \boldsymbol{c}^{\boldsymbol{\chi}}_i[1 - \text{Mask}^{\boldsymbol{\chi}}(\Psi^{\mathcal{S}}))] \leftarrow 1, \boldsymbol{\theta}^{\boldsymbol{\chi}} \leftarrow \{\boldsymbol{m}^{\boldsymbol{\chi}}_i, \boldsymbol{c}^{\boldsymbol{\chi}}_i\}$
$\boldsymbol{m}^{\psi}_i \sim p^{\psi}_F(\boldsymbol{m}^{\psi}_i | \psi; \alpha^{\psi}_{1:i}), \boldsymbol{c}^{\psi}_i \sim p^{\psi}_F(\boldsymbol{c}^{\psi}_i | \psi; \alpha^{\psi}_{1:i}), \boldsymbol{\theta}^{\psi} \leftarrow \{\boldsymbol{m}^{\psi}_i, \boldsymbol{c}^{\psi}_i\}$
feed all protein modalities into the network
$\boldsymbol{\theta}^{\mathcal{P}} \leftarrow \{\boldsymbol{\theta}^{\mathcal{S}}, \boldsymbol{\theta}^{\boldsymbol{p}}, \boldsymbol{\theta}^{\boldsymbol{R}}, \boldsymbol{\theta}^{\psi}, \boldsymbol{\theta}^{\boldsymbol{\chi}}\}$
$\Psi^{\mathcal{S}}, \Psi^{\boldsymbol{p}}, \Psi^{\boldsymbol{R}}, \Psi^{\psi}, \Psi^{\boldsymbol{\chi}} \leftarrow \Psi(\boldsymbol{\theta}^{\mathcal{P}}, t)$
calculate losses
$\mathcal{L}_{\boldsymbol{q}} \leftarrow n\, \alpha^{\boldsymbol{q}}_i \frac{I_{\nu+1}(\alpha^{\boldsymbol{q}}_i)}{I_{\nu}(\alpha^{\boldsymbol{q}}_i)}(1 - \text{dot}(\boldsymbol{q}, \Psi^{\boldsymbol{q}}(\boldsymbol{\theta}^{\mathcal{P}}, i)))$
$\mathcal{L}_{\mathcal{S}} \leftarrow n\, \mathbb{E}_{\mathcal{N}(\mathbf{y} | \alpha_i (K \mathbf{e}_{\mathcal{S}} - \mathbf{1}), \alpha_i K \boldsymbol{I})} \ln \mathcal{N}\left(\mathbf{y} \mid \alpha_i \left(K \mathbf{e}_{\mathcal{S}} - \mathbf{1}\right), \alpha_i K \boldsymbol{I}\right)$
$\qquad - \sum_{d=1}^{L} \ln \left(\sum_{k=1}^{K} p_O^{(d)}(k \mid \boldsymbol{\theta}^{\mathcal{S}}; t_{i-1}) \mathcal{N}\left(y^{(d)} \mid \alpha_i \left(K \mathbf{e}_k - \mathbf{1}\right), \alpha_i K \boldsymbol{I}\right)\right)$
$\mathcal{L}_{\boldsymbol{p}} \leftarrow \frac{n}{2}\left(1 - \sigma_1^{2/n}\right) \frac{\left\| \boldsymbol{p} - \Psi^{\boldsymbol{p}}_{i-1}(\boldsymbol{\theta}^{\mathcal{P}}, t) \right\|^2}{\sigma_1^{2i/n}}$
$\mathcal{L}_{\boldsymbol{\chi}} \leftarrow n\, \alpha^{\boldsymbol{\chi}}_i \frac{I_1(\alpha^{\boldsymbol{\chi}}_i)}{I_0(\alpha^{\boldsymbol{\chi}}_i)}(1 - \cos(\boldsymbol{\chi} - \Psi^{\boldsymbol{\chi}}(\boldsymbol{\theta}^{\mathcal{P}}_{i-1}, t_{i-1})))$
$\mathcal{L}_{\boldsymbol{\chi}} \leftarrow \mathcal{L}_{\boldsymbol{\chi}} \cdot \text{Mask}^{\boldsymbol{\chi}}(\mathcal{S})$
$\mathcal{L}_{\psi} \leftarrow n\, \alpha^{\psi}_i \frac{I_1(\alpha^{\psi}_i)}{I_0(\alpha^{\psi}_i)}(1 - \cos(\psi - \Psi^{\psi}(\boldsymbol{\theta}^{\mathcal{P}}_{i-1}, t_{i-1})))$
Optimize $\lambda_{\boldsymbol{q}} \mathcal{L}_{\boldsymbol{q}} + \lambda_{\mathcal{S}} \mathcal{L}_{\mathcal{S}} + \lambda_{\boldsymbol{p}} \mathcal{L}_{\boldsymbol{p}} + \lambda_{\boldsymbol{\chi}} \mathcal{L}_{\boldsymbol{\chi}} + \lambda_{\psi} \mathcal{L}_{\psi}$

Algorithm 2 Sampling

Require: $n, K \in \mathbb{Z}, \sigma_1 \in \mathbb{R}^+, \beta_1 \in \mathbb{R}^+, \alpha^{\boldsymbol{\chi}}_{1:n}, \alpha^{\psi}_{1:n}, \alpha^{\boldsymbol{q}}_{1:n} \in \mathbb{R}^+$
initialize the prior parameters
$\boldsymbol{\mu}^{\boldsymbol{q}} \sim U_{\mathbb{S}^3}, \boldsymbol{\kappa}^{\boldsymbol{q}} \leftarrow \mathbf{0}, \boldsymbol{\theta}^{\boldsymbol{q}} \leftarrow \{\boldsymbol{\mu}^{\boldsymbol{q}}, \boldsymbol{\kappa}^{\boldsymbol{q}}\}; \quad \boldsymbol{\theta}^{\mathcal{S}}_0 \leftarrow \frac{1}{K}\mathbf{1}_{1 \times L}; \quad \boldsymbol{\mu}^{\boldsymbol{p}} \leftarrow \mathbf{0}, \boldsymbol{\theta}^{\boldsymbol{p}} \leftarrow \{\boldsymbol{\mu}^{\boldsymbol{p}}\}$
$\boldsymbol{m}^{\boldsymbol{\chi}} \leftarrow U(0, 2\pi), \boldsymbol{c}^{\boldsymbol{\chi}} \leftarrow \mathbf{0}, \boldsymbol{\theta}^{\boldsymbol{\chi}} \leftarrow \{\boldsymbol{m}^{\boldsymbol{\chi}}, \boldsymbol{c}^{\boldsymbol{\chi}}\}; \quad \boldsymbol{m}^{\psi} \leftarrow U(0, 2\pi), \boldsymbol{c}^{\psi} \leftarrow \mathbf{0}, \boldsymbol{\theta}^{\boldsymbol{\chi}} \leftarrow \{\boldsymbol{m}^{\psi}, \boldsymbol{c}^{\psi}\}$
for $i \leftarrow 1, \cdots, n; t \leftarrow \frac{i-1}{n}$ **do**
  # use network to do inter-dependency modeling across dimensions of all modalities
  $\boldsymbol{\theta}^{\mathcal{P}} \leftarrow \{\boldsymbol{\theta}^{\mathcal{S}}, \boldsymbol{\theta}^{\boldsymbol{p}}, \boldsymbol{\theta}^{\boldsymbol{R}}, \boldsymbol{\theta}^{\psi}, \boldsymbol{\theta}^{\boldsymbol{\chi}}\}$
  $\Psi^{\mathcal{S}}, \Psi^{\boldsymbol{p}}, \Psi^{\boldsymbol{R}}, \Psi^{\psi}, \Psi^{\boldsymbol{\chi}} \leftarrow \Psi(\boldsymbol{\theta}^{\mathcal{P}}, t)$
  if $i = n$ **then**
    break
  end if
  # update each modality with Bayesian flow
  $\boldsymbol{\theta}^{\boldsymbol{q}} \sim p^{\boldsymbol{q}}_F(\boldsymbol{\theta}^{\boldsymbol{q}}_i | \Psi^{\boldsymbol{R}}; \alpha^{\boldsymbol{q}}_{1:i})$
  $\boldsymbol{\theta}^{\mathcal{S}} \sim p^{\mathcal{S}}_F(\boldsymbol{\theta}^{\mathcal{S}} \mid \Psi^{\mathcal{S}}; t)$
  $\boldsymbol{\theta}^{\boldsymbol{p}} \leftarrow \boldsymbol{\mu}^{\boldsymbol{p}} \sim p^{\boldsymbol{p}}_F(\boldsymbol{\mu}^{\boldsymbol{p}} | \boldsymbol{p}; t)$
  $\boldsymbol{m}^{\boldsymbol{\chi}}_i \sim p^{\boldsymbol{\chi}}_F(\boldsymbol{m}^{\boldsymbol{\chi}}_i | \boldsymbol{\chi}; \alpha^{\boldsymbol{\chi}}_{1:i}), \boldsymbol{c}^{\boldsymbol{\chi}}_i \sim p^{\boldsymbol{\chi}}_F(\boldsymbol{c}^{\boldsymbol{\chi}}_i | \boldsymbol{\chi}; \alpha^{\boldsymbol{\chi}}_{1:i})$,
  $\boldsymbol{m}^{\boldsymbol{\chi}}_i \leftarrow \boldsymbol{m}^{\boldsymbol{\chi}}_i \cdot \text{Mask}^{\boldsymbol{\chi}}(\Psi^{\mathcal{S}}), \boldsymbol{c}^{\boldsymbol{\chi}}_i[1 - \text{Mask}^{\boldsymbol{\chi}}(\Psi^{\mathcal{S}}))] \leftarrow 1, \boldsymbol{\theta}^{\boldsymbol{\chi}} \leftarrow \{\boldsymbol{m}^{\boldsymbol{\chi}}_i, \boldsymbol{c}^{\boldsymbol{\chi}}_i\}$
  $\boldsymbol{m}^{\psi}_i \sim p^{\psi}_F(\boldsymbol{m}^{\psi}_i | \psi; \alpha^{\psi}_{1:i}), \boldsymbol{c}^{\psi}_i \sim p^{\psi}_F(\boldsymbol{c}^{\psi}_i | \psi; \alpha^{\psi}_{1:i}), \boldsymbol{\theta}^{\psi} \leftarrow \{\boldsymbol{m}^{\psi}_i, \boldsymbol{c}^{\psi}_i\}$
end for
Return $\Psi^{\mathcal{S}}, \Psi^{\boldsymbol{p}}, \Psi^{\boldsymbol{R}}, \Psi^{\psi}, \Psi^{\boldsymbol{\chi}}$

E Limitations and broader impacts

Limitations Despite the promising results of ProBayes, we acknowledge that the binding affinities and complex stability assessed by Rosetta serve as an initial screening step. Real-world properties of the generated candidates still require validation through costly wet-lab experiments. However, this challenge is shared across the research community.

Broader impacts Our work contributes to the advancement of unified, end-to-end protein design, potentially enabling more accurate generation of functional proteins. This has implications for therapeutic discovery, enzyme engineering, and synthetic biology. However, as with any generative biological technology, careful oversight is essential to prevent unintended misuse, such as the design of harmful or dual-use proteins.

F Experiment details

Algorithm 3 Invariant point attention with quaternion implementation (IPAq)

Require: $\{\boldsymbol{s}_i\}, \{\boldsymbol{z}_{ij}\}, \{T_i^{\mathcal{Q}}\}, N_{\text{head}} = 12, c = 16, N_{\text{query points}} = 4, N_{\text{point values}} = 8$

$\boldsymbol{q}_i^h, \boldsymbol{k}_i^h, \boldsymbol{v}_i^h = \text{LinearNoBias}(s_i)$ $\quad \{\boldsymbol{q}_i^h, \boldsymbol{k}_i^h, \boldsymbol{v}_i^h \in \mathbb{R}^c, h \in \{1, \dots, N_{\text{head}}\}\}$

$\boldsymbol{q}_i^{hp}, \boldsymbol{k}_i^{hp} = \text{LinearNoBias}(s_i)$ $\quad \boldsymbol{q}_i^{hp}, \boldsymbol{k}_i^{hp} \in \mathbb{R}^3, p \in \{1, \dots, N_{\text{query points}}\}$

$\boldsymbol{v}_i^{hp} = \text{LinearNoBias}(s_i)$ $\quad \boldsymbol{v}_i^{hp} \in \mathbb{R}^3, p \in \{1, \dots, N_{\text{point values}}\}$

$\boldsymbol{b}_{ij}^h = \text{LinearNoBias}(z_{ij})$

$w_C = \sqrt{\frac{2}{9N_{\text{query points}}}}$

$w_L = \sqrt{\frac{1}{3}}$

$a_{ij}^h = \text{softmax}_j\left(w_L\left(\frac{1}{\sqrt{c}}\boldsymbol{q}_i^{h\top}\boldsymbol{k}_j^h + \boldsymbol{b}_{ij}^h - \frac{\gamma^h w_C}{2}\sum_p \left\| T_i^{\mathcal{Q}} \circ \boldsymbol{q}_i^{hp} - T_j^{\mathcal{Q}} \circ \boldsymbol{k}_j^{hp}\right\|^2\right)\right)$

$\tilde{\boldsymbol{o}}_i^h = \sum_j a_{ij}^h \boldsymbol{z}_{ij}$

$\boldsymbol{o}_i^h = \sum_j a_{ij}^h \boldsymbol{v}_j^h$

$\boldsymbol{o}_i^{hp} = (T_i^{\mathcal{Q}})^{-1} \circ \sum_j a_{ij}^h \left(T_j^{\mathcal{Q}} \circ \boldsymbol{v}_j^{hp}\right)$

$\tilde{\boldsymbol{s}}_i = \text{Linear}\left(\text{concat}_{h,p}(\tilde{\boldsymbol{o}}_i^h, \boldsymbol{o}_i^h, \boldsymbol{o}_i^{hp}, \|\boldsymbol{o}_i^{hp}\|)\right)$

return $\{\tilde{\boldsymbol{s}}_i\}$

We provide our code in https://github.com/GenSI-THUAIR/ProBayes.

F.1 Rationalized information flow implementation

In fact, the rationalized information flow can be implemented in various ways using different parameter sizes or by mixing modules, with the only constraint being the input and output specifications of each module. Here we illustrate our instantiations for each module. Firstly, we explain the key component, *i.e.*, IPA with quaternion implementation.

Invariant point attention with quaternion implementation We illustrate the invariant point attention [Jumper et al., 2021] implemented by the unit quaternion in Algorithm 3 using the orange color to denote the modification compared to the original IPA. Specifically, we only replace the original implementation of the SE(3) transformation based on rotation matrix and translation $T_i = \{\boldsymbol{R}, \boldsymbol{t}\}$ with the quaternion-based one $T_i^{\mathcal{Q}} = \{\boldsymbol{q}, \boldsymbol{t}\}$. For a point with homogeneous representation $\boldsymbol{p} = [x, y, z, 0]^T$ and $T_i^{\mathcal{Q}}$, the transformation is:

$$T_i^{\mathcal{Q}} \circ \boldsymbol{p} = \boldsymbol{q} \times \boldsymbol{p} \times (\boldsymbol{q})^{-1} + \boldsymbol{t} \tag{36}$$

where $\times$ is the Hamilton product for two quaternions [Hazewinkel et al., 2006] and the inverse transformation $(T_i^{\mathcal{Q}})^{-1} = \{(\boldsymbol{q})^{-1}, -\boldsymbol{t}\}$. Such operations are antipodal invariant because $\boldsymbol{q} \times \boldsymbol{p} \times (\boldsymbol{q})^{-1} = (-\boldsymbol{q}) \times \boldsymbol{p} \times (-\boldsymbol{q})^{-1}$. Therefore, the IPAq module defined in Algorithm 3 is an antipodal invariant transformation.

`Sequence&Backbone Mixing Module` **implementation** The implementation of this module builds upon the graph attention-based architecture `FramePred` [Yim et al., 2023b], with the following modifications: 1) The initial node embedding is mixed with the noised sequence $\boldsymbol{\theta}^{\mathcal{S}}$, the noised backbone torsion angles $\boldsymbol{\theta}^{\psi}$, and the timestep t via an MLP as follows:

$$\mathbf{h}_0' = \text{MLP}(\mathbf{h}_0, \text{embed}^{\mathcal{S}}(\boldsymbol{\theta}^{\mathcal{S}}), \text{embed}^{\psi}(\boldsymbol{\theta}^{\psi}), \text{embed}^{t}(t)) \tag{37}$$

where $\text{embed}^{\mathcal{S}}$, embed^{ψ}, embed^{t} are embedders for sequence, angles, and time, respectively. 2) The IPA module is modified as stated above and also in Algorithm 3. 3) The prediction of ψ is removed and repositioned at the end of the Backbone & Side-chain Mixing module.

`Backbone&Sidechain Mixing Module` **implementation** To capture all-atom geometry during backbone prediction, we introduce an additional multilayer perceptron (MLP) that embeds sidechain features. This sidechain embedding is then passed to the `BackboneUpdate` module [Jumper et al., 2021], enabling refined updates to the backbone frame. The sidechain embedding module comprises three linear layers with bias terms, followed by a LayerNorm layer. Conversely, to incorporate backbone information into sidechain design, we employ another MLP-based module consisting of linear layers with bias terms—which jointly encodes the node embeddings from the `Sequence&Backbone Mixing Module` alongside the sidechain features.

F.2 Peptide design

Metrics Consistent with previous research [Kong et al., 2024], we generate 40 candidate structures per receptor for the sequence-structure co-design task and 10 candidates for each receptor-ligand pair in the binding conformation generation task. The evaluation metrics are detailed as follows:

ΔG [Kong et al., 2024] The binding energy (in kcal/mol) calculated by Rosetta [Alford et al., 2017] to evaluate the binding affinity of the generated peptide using the function `InterfaceAnalyzer`.

Success rate [Kong et al., 2024] The proportion of successful designs i.e., those with $\Delta G < 0$, out of all generated candidates.

DockQ [Basu and Wallner, 2016] A comprehensive metric that evaluates the all-atom similarity at the interface between a candidate and the reference complex.

RMSD$_{\text{C}_\alpha}$ [Kong et al., 2024] The root mean square deviation (RMSD) of the C_α coordinates between a candidate and the reference structure, measured in Ångströms (Å).

Validity [Lin et al., 2024] refers to the proportion of peptides that are chemically valid, determined by whether the generated atomic bond lengths are within 0.5A above and below the ideal value.

Diversity [Lin et al., 2024] is measured by the average pairwise distance of the generated peptides using TM score and sequence similarity.

Novelty [Lin et al., 2024] A generated peptide is considered novel if both its TM-score and sequence overlap with the reference are both below 0.5.

Note that the diversity and novelty are evaluated combining the validity to avoid false positives.

RMSD$_{\text{atom}}$ [Kong et al., 2024] on all atoms to measure the quality of the all-atom geometry for the binding conformation generation task.

Baselines For RFDiffusion, we set the number of cycles to one and disable the empirical force field refinement to ensure a fair comparison with other methods. For PepFlow, we use the official implementations and retrain the models on the same datasets using the default hyperparameters provided in its repository. For PepGLAD, most evaluation metrics are taken directly from the original paper, while the design metrics are computed using the official checkpoint available in its repository.

F.3 Antibody design

Following Ye et al. [2024], we generate 64 candidates for each receptor in the sequence-structure co-design task and report the average performance across these candidates.

Metrics The metrics for antibody design include:

AAR [Ye et al., 2024] The amino acid recovery rate, calculated as the number of residues in the generated CDR-H3 sequences that match the reference antibody;

RMSD [Ye et al., 2024] The root-mean-square deviation, measured between the generated and natural antibodies using the Cα coordinates of the CDR-H3 region.

E$_{\text{total}}$ [Ye et al., 2024] The total energy computed using Rosetta's full-atom score function with the default REF15 weight set, evaluated on the CDR-H3 regions.

ΔG [Ye et al., 2024] The binding energy, analogous to that used in peptide design, but here evaluated between the CDR-H3 region and the antigen.

Baselines We directly borrow the baseline results reported in [Ye et al., 2024].

F.4 Other implementation details

Computation resources All experiments in this paper are conducted on a node with 8 NVIDIA A100 80GB. Each training task requires 4 GPUs.

Relaxation For peptide design, we adopt the relaxation procedure described in Kong et al. [2024], which optimizes the full all-atom structure. In contrast, for antibody design, we employ the protocol from Ye et al. [2024], which performs relaxation exclusively on side-chain atoms while keeping the backbone atoms fixed.

Hyperparameters For all experiments, the network is configured with a node embedding size of 128 and an edge embedding size of 64. Node embeddings in the IPAq module have 128 dimensions, and edge embeddings have 64 dimensions. The IPAq attention mechanism comprises 8 heads, with 8 query-key points and 12 value points for geometric attention. Additionally, a sequence transformer is integrated into the encoder, featuring 4 attention heads and 2 layers to enhance contextual representation. The entire encoder architecture consists of 6 stacked IPAq blocks, enabling deep and expressive modeling of the input molecular graph structure. Finally, the network consists of 7.02 million parameters. For optimization, we set the learning rate to 5×10^{-4} and use a batch size of 40 per distributed node with Adam optimizer.

G Comparison between BFN and SDEs/ODEs

A recent study by Xue et al. [2024] explores potential connections between continuous-time Bayesian Flow Networks and SDEs. However, we clarify that the BFNs are fundamentally distinct from SDE/ODE with evidence as follows:

1. The Bayesian flow distribution, as defined in Graves et al. [2023], does not involve differential terms, which are central to the formulation of SDEs and ODEs.
2. The Bayesian flow formulation employed in this work is *discrete-time* and does not rely on any discretization of continuous-time dynamics, in contrast to SDEs- and ODEs-based approaches.
3. In both the hypertorus Bayesian flows [Wu et al., 2025] and the hypersphere Bayesian flows introduced in this paper, the accumulated accuracy parameters κ_i are inherently stochastic. These stochastic components cannot be fully represented or captured within a conventional SDE framework.
4. The connection proposed in Xue et al. [2024] is not mathematically rigorous: it introduces a truncation of the time domain from $[0, 1]$ to $[0, 1 - \eta]$, where η is a parameter, thereby deviating from the standard continuous-time SDE formulation.

H More results

H.1 Error bars

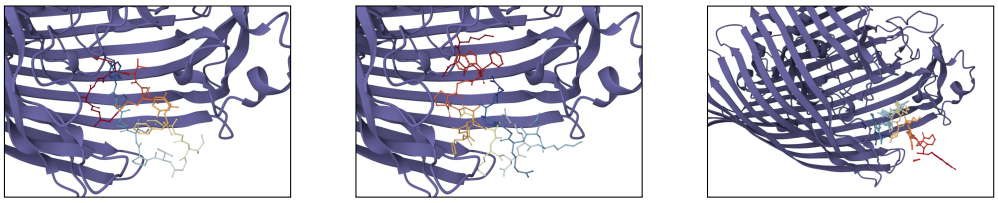
Table 6: Peptide design task results with error bars representing standard deviations from three independent experiments using distinct random seeds.

Dataset	Method	Energy		Native Likeness		Design		
		$\Delta G(\downarrow)$	Success(↑)	DockQ(↑)	$\text{RMSD}_{C_\alpha}(\downarrow)$	Valid (↑)	V&Div(↑)	V&Novel(↑)
PepBench	ProBayes	-28.63±0.39	72.92±0.79	0.740±0.037	2.28±0.036	0.998±0.00012	0.449±0.0021	0.728±0.0014

We report the error bars for the protein design task, as shown in Tab. 6, based on three independent experiments using different random seeds.

H.2 Visualization

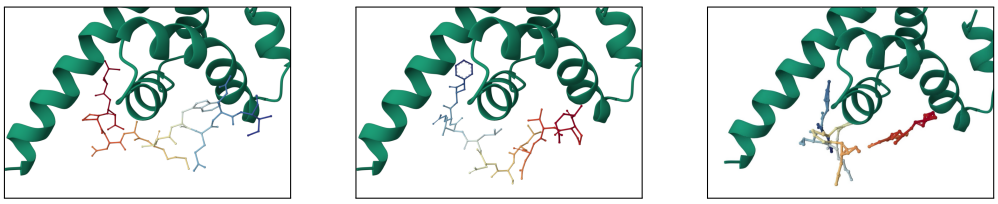
We provide the visualization of designed peptides and antibodies in Fig. 8 and Fig. 9.



(a) Reference: ΔG = -34.21, PDB=4cu4

(b) Generated: ΔG = -39.42, PDB=4cu4

(c) PepGLAD: ΔG = -30.71, PDB=4cu4

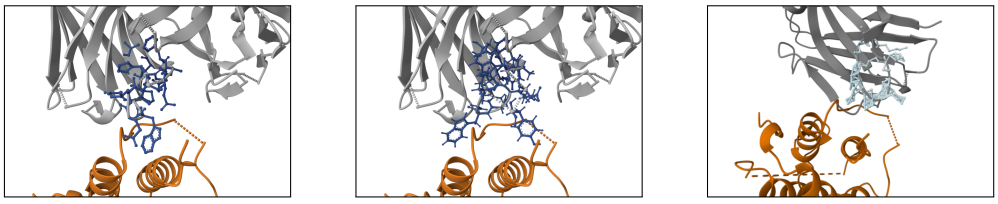


(d) Reference: ΔG = -33.32, PDB=3pkn

(e) Generated: ΔG = -33.88, PDB=3pkn

(f) PepGLAD: ΔG = -13.39, PDB=3pkn

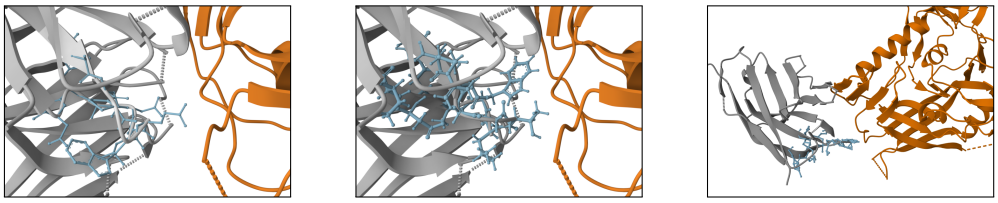
Figure 8: Visualization of designed peptides.



(a) Reference: ΔG = -18.76, PDB=5j13

(b) Generated: ΔG = -23.64, PDB=5j13

(c) dyMEAN: ΔG = +23911.25, PDB=5j13



(d) Reference: ΔG = -0.45, PDB=4dvr

(e) Generated: ΔG = -7.42, PDB=4dvr

(f) dyMEAN: ΔG = +11654.77, PDB=4dvr

Figure 9: Visualization of designed antibodies.

I Further discussion of information shortcut

Protpardelle [Chu et al., 2024] tries to remedy the information leakage by noising all 37 unique atom position inputs instead of only the atoms corresponding to the sequence. However, we have proved in the main text that even noisy side-chain information can create an information shortcut for the sequence prediction. Therefore, the information shortcut still exists in Protpardelle.

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: Our main experiment and ablation study support our claims.

   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: We discuss the limitations of this work in Appendix E.

   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]

Justification: We provide the proof of the propositions in Appendix A.

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: We provide our implementation details in Appendix F

   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: [Yes]

Justification: We have provided our code link.

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: We provide the training and test details in Appendix F.

   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: We provide the error bars in Appendix H.

   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).
   - 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: We provide this in Appendix F.

   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: This research conforms with the NeurIPS Code of Ethics in every respect.

   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: We provide the discussion of broader impacts in Appendix E.

    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: [Yes]

    Justification: We provide the discussion in Appendix E.

    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: We provide this in Appendix F.

    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.
    - 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: [Yes]

Justification: We provide the documentation for our code.

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: This 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: [Yes]

Justification: We provide the discussion of this in Appendix E.

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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.