Fine-tuning BioEmu for Accurate Protein Folding Stability Prediction

Zhaoyang Li [*1]

Abstract

Recent advances in protein structure prediction models such as AlphaFold have largely resolved static folding problems. However, accurately profiling dynamic protein conformations to estimate folding stabilities remains a huge challenge. Biomolecular Emulator (BioEmu) is a recent generative deep learning framework that employs a Property Prediction Fine-Tuning (PPFT) algorithm that integrates extensive MEGAScale experimental datasets with molecular dynamics (MD) simulations to infer folding free energies. Despite its innovative design, preliminary fine-tuning results revealed limitations in predictive accuracy. In this work, we propose a novel approach that efficiently fine-tunes a $\mathrm{SE}(3)$ equivariant diffusion model using experimental expectation values, while preserving the majority of the pretrained parameters to maintain the integrity of the underlying diffusion process. This work may mark a significant advance in the integration of experimental data with deep generative models, paving the way for more reliable computational assessments of protein folding energy landscapes.

1. Introduction

Predicting protein stability (e.g. folding free energy changes $\Delta\Delta G$) from sequence and limited structural information is a long-standing challenge. Biomolecular Emulator (BioEmu) is a recently proposed generative diffusion model that addresses this by emulating protein conformational ensembles and integrating experimental stability data (Lewis et al., 2025). In BioEmu's original framework, a property-prediction fine-tuning (PPFT) algorithm was used to incorporate experimental stability measurements without requiring known structures. PPFT works by generating a small ensemble of structures (using a fast 8-step diffusion sampling) and comparing an observable (fraction of folded structures) to experimental values, then backpropagating the error to adjust the model. This strategy enabled BioEmu to predict stability with high accuracy, outperforming black-box sequence-based models.

However, the PPFT approach has limitations: it introduces an approximation in sampling that may perturb the pretrained distribution, and it does not explicitly account for the geometrical symmetries of protein conformations. Recent advances in $\mathrm{SE}(3)$ equivariant diffusion modeling provide a more principled framework for generative processes on the manifold of rigid-body transformations (Yim et al., 2023). Our central idea is to reinterpret fine-tuning as a constrained optimization on the manifold of protein conformations, where we impose that certain expected observables match experimental values while minimally perturbing the pretrained ensemble distribution. This project will pursue that idea in four stages, each serving as an independent milestone:

2. Preliminaries and Notation

Throughout this paper we adopt the following notation and conventions.

Manifolds and Lie groups. Let $\mathcal{M}$ denote a smooth, d-dimensional Riemannian manifold with metric $\langle\cdot,\cdot\rangle_{\mathcal{M}}$ and associated volume form $\mathrm{d}V$. $\mathbf{X} \in \mathcal{M}$ is a point on the manifold. We write $\mathrm{SO}(3)$ for the group of 3×3 rotation matrices and $\mathfrak{so}(3)$ for its Lie algebra, and similarly $\mathrm{SE}(3) \cong \mathrm{SO}(3) \ltimes \mathbb{R}^3$ with Lie algebra $\mathfrak{se}(3) = \mathfrak{so}(3) \oplus \mathbb{R}^3$.

For any Lie group G and its Lie algebra $\mathfrak{g}$, $\exp : \mathfrak{g} \to G$ is the Riemannian exponential map, and $\log : G \to \mathfrak{g}$ its (local) inverse. The isomorphism hat $[\cdot]^\wedge : \mathbb{R}^d \to \mathfrak{g}$ and vee $[\cdot]^\vee : \mathfrak{g} \to \mathbb{R}^d$, i.e. the vectorization and de-vectorization maps, induce $\mathrm{Exp} : \mathbb{R}^d \to G$ and $\mathrm{Log} : G \to \mathbb{R}^d$ by the composition of mappings, respectively.

The left action of $g \in G$ on $h \in G$ is $L_g(h) = gh$ and its differential is $\mathrm{d}L_g : T_hG \to T_{gh}G$. The metric on $\mathrm{SE}(3)$ is given by the canonical left-invariant metric, which is induced by the standard inner product on $\mathfrak{so}(3)$ and $\mathbb{R}^3$, i.e. $\langle \mathbf{t}_1, \mathbf{t}_2 \rangle_{\mathrm{SE}(3)} = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle_{\mathbb{R}^3} + \langle \mathbf{r}_1, \mathbf{r}_2 \rangle_{\mathrm{SO}(3)}$ for $\mathbf{t}_1 = (\mathbf{r}_1, \mathbf{x}_1) \in \mathfrak{se}(3)$ and $\mathbf{t}_2 = (\mathbf{r}_2, \mathbf{x}_2) \in \mathfrak{se}(3)$. The bi-

[*]Equal contribution [1]Department of Bioengineering, Stanford University, CA 94305, USA. Correspondence to: Zhaoyang Li <zhaoyangli@stanford.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

invariant Riemannian metric on $\mathrm{SO}(3)$ is given by its Killing form $\langle \mathbf{r}_1, \mathbf{r}_2 \rangle_{\mathrm{SO}(3)} = \frac{1}{2}\mathrm{tr}(\mathbf{r}_1^{\mathrm{T}} \mathbf{r}_2)$, where $\mathbf{r}_1, \mathbf{r}_2 \in \mathfrak{so}(3)$.

Protein backbone frames. A protein backbone of N residues is represented by a sequence of rigid frames

$$\mathbf{T} = (\mathbf{T}_1, \ldots, \mathbf{T}_N) \in \mathrm{SE}(3)^N, \quad \mathbf{T}_n = (\mathbf{R}_n, \mathbf{x}_n)$$

where $\mathbf{R}_n \in \mathrm{SO}(3)$ is the rotation matrix and $\mathbf{x}_n \in \mathbb{R}^3$ is the translation vector for residue n. Each frame acts on the idealized residue coordinates $(\mathrm{N}^*, \mathrm{C}^*, \mathrm{C}_\alpha^*) \subset \mathbb{R}^3$ via $\mathbf{T}_n(v) = \mathbf{R}_n v + \mathbf{x}_n$, so that the atomic positions for residue n are

$$(\mathrm{N}_n, \mathrm{C}_n, \mathrm{C}_{\alpha n}) = \mathbf{T}_n(\mathrm{N}^*, \mathrm{C}^*, \mathrm{C}_\alpha^*),$$

and the O atom is placed by an additional torsion angle ψ_n around the $\mathrm{C}_\alpha - \mathrm{C}$ bond.

Diffusion on manifolds. We consider time $t \in [0, 1]$ and a forward and reverse Itô SDE on $\mathcal{M}$:

$$\mathrm{d}\mathbf{X}_t = b(\mathbf{X}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t^{\mathcal{M}} \tag{1}$$

$$\mathrm{d}\mathbf{X}_t = \big(b(\mathbf{X}_t, t) - g(t)^2 \nabla_{\mathbf{X}_t} \log p_t(\mathbf{X}_t)\big)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t^{\mathcal{M}} \tag{2}$$

where $\mathbf{W}_t^{\mathcal{M}}$ is Brownian motion on $\mathcal{M}$, $b(\mathbf{X}_t, t)$ a drift term, and $g(t)$ a diffusion coefficient. The time-reversed process requires the Stein score $\nabla_{\mathbf{X}_t} \log p_t(\mathbf{X}_t)$, which is the Riemannian gradient of the log-density.

Distributions on Lie groups. We denote the isotropic Gaussian distribution on $\mathrm{SO}(3)$ as $\mathrm{IGSO}(3)(\mathbf{R}_0, \sigma^2)$, where $\mathbf{R}_0$ is the mean rotation and σ^2 is the variance. The probability density function (PDF) of $\mathbf{R}_t \sim \mathrm{IGSO}(3)(\mathbf{R}_0, \sigma^2)$ when $t = \sigma^2$ is given by

$$p(\mathbf{R}_t; \mathbf{R}_0, \sigma) = \frac{1}{8\pi^2} \sum_{\ell=0}^{\infty} (2\ell + 1) e^{-\frac{\sigma^2}{2}\ell(\ell+1)} \chi_\ell(\mathbf{R}_0^{\mathrm{T}} \mathbf{R}_t) \tag{3}$$

with respect to the canonical Haar measure on $\mathrm{SO}(3)$, $\mu_{\mathrm{SO}(3)} = 4\sin^2\frac{\omega}{2}\mathrm{d}\omega \wedge \mathrm{d}\Omega$. Here χ_ℓ is the ℓ-th irreducible unitary representation of dimension $2\ell+1$ and Ω is the solid angle on $\mathbb{S}^2$. The axis-angle representation $\mathbf{q} = \mathrm{Log}(\mathbf{R})$ and $\omega = \|\mathbf{q}\|_2$ is used to describe the rotation for score matching. A random variable $\mathbf{R}_t \sim \mathrm{IGSO}(3)(\mathbf{R}_0, \sigma^2)$ is sampled from $\mathbf{R}_0\,\mathrm{IGSO}(3)(\mathbf{I}, \sigma^2)$, where $\mathbf{I}$ is the identity matrix.

3. Stage 1: Prototype $\mathrm{IGSO}(3)$ Diffusion on $\mathrm{SO}(3)$

The first milestone is to prototype an isotropic Gaussian $\mathrm{SO}(3)$ ($\mathrm{IGSO}(3)$) diffusion process using a toy problem (Leach et al., 2022). We will begin with a simple synthetic distribution on $\mathrm{SO}(3)$ (for example, a bimodal mixture of two distinct orientations) and attempt to sample from it using an $\mathrm{IGSO}(3)$ diffusion process. Key steps include: (i) formulating the forward and reverse diffusion SDEs for $\mathbf{X} \in \mathcal{M}$ (Bortoli et al., 2022); (ii) parameterizing $\mathrm{SO}(3)$ diffusion with a suitable representation of rotation (Solà et al., 2021);

Proposition 3.1 (Marginal Distribution of $\mathrm{IGSO}(3)$)**.** *Let $\mathbf{R}_t \sim \mathrm{IGSO}(3)(\mathbf{R}_0, \sigma^2)$ be a random rotation matrix. The marginal distribution of its rotation angle $\omega_t = \|\mathrm{Log}(\mathbf{R}_t)\|_2$ is given by its PDF $\frac{1-\cos\omega_t}{\pi} f(\omega_t; \mathbf{R}_0, \sigma)$, where $f(\omega_t; \mathbf{R}_0, \sigma)$ is defined as*

$$f(\omega_t; \mathbf{R}_0, \sigma) = \sum_{\ell=0}^{\infty} e^{-\frac{\sigma^2}{2}\ell(\ell+1)} \frac{\sin\left(\ell + \frac{1}{2}\right)\omega_0}{\sin\frac{\omega_0}{2}} \frac{\sin\left(\ell + \frac{1}{2}\right)\omega_t}{\sin\frac{\omega_t}{2}} \tag{4}$$

Here $\omega_0 = \|\mathrm{Log}(\mathbf{R}_0)\|_2$ is the rotation angle of $\mathbf{R}_0$.

Proposition 3.2 (Axis-Angle Decomposition of the $\mathrm{IGSO}(3)$ Distribution)**.** *Let $\mathbf{R} = \mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t$ be a random rotation matrix sampled from $\mathrm{IGSO}(3)(\mathbf{I}, \sigma^2)$ (so that $\mathbf{R}_t \sim \mathrm{IGSO}(3)(\mathbf{R}_0, \sigma^2)$). Let $\mathbf{q} = \mathrm{Log}(\mathbf{R})$ and $\omega = \|\mathbf{q}\|_2$ be the axis-angle representation. Then the axis-angle decomposition of $\mathbf{R}$ is given by*

$$\frac{\mathbf{q}}{\omega} \sim \mathcal{U}(\mathbb{S}^2) \tag{5}$$

$$\omega \sim \frac{1 - \cos\omega}{\pi} f(\omega; \mathbf{I}, \sigma) \tag{6}$$

where $\mathcal{U}(\mathbb{S}^2)$ is the uniform distribution on the unit sphere $\mathbb{S}^2$.

and (iii) training a neural network $s_\theta(\mathbf{R}_t, t) \in \mathbb{R}^3$ to approximate the score function via denoising score matching (Song et al., 2021).

Proposition 3.3 (Form of the Stein Score on $\mathrm{SO}(3)$)**.** *Let $\mathbf{R}_0, \mathbf{R}_t \in \mathrm{SO}(3)$ and write their relative rotation as $\mathbf{R} = \mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t$. Then the Stein score function $s^*(\mathbf{q}, t) \in \mathbb{R}^3$ at time $t = \sigma^2$ of the reverse diffusion process satisfies*

$$\begin{aligned} s^*(\mathbf{q}, t) &= \left[\mathbf{R}_t^{\mathrm{T}} \nabla_{\mathbf{R}_t} \log p_t(\mathbf{R}_t \mid \mathbf{R}_0)\right]^\vee \\ &= \frac{\mathbf{q}}{\omega} \frac{\partial}{\partial\omega} \log f(\omega; \mathbf{I}, \sigma) \end{aligned} \tag{7}$$

where $p_t(\mathbf{R}_t \mid \mathbf{R}_0)$ is the conditional distribution of $\mathbf{R}_t$ given $\mathbf{R}_0$ and $\omega = \|\mathrm{Log}(\mathbf{R})\|_2$ is the relative rotation angle.

The training objective is to minimize the denoising score matching loss:

$$\arg\min_\theta \mathbb{E}_{\mathbf{R}_0, \mathbf{R}_t \mid \mathbf{R}_0, t}\left[\|\lambda(t) s_\theta(\mathbf{R}_t, t) - \lambda(t) s^*(\mathbf{q}, t)\|_2^2\right] \tag{8}$$

where $\lambda(t)$ is a time-dependent weighting function.

We will visualize the rotation angle distribution to confirm that the diffusion model can recover the mixture. Successfully generating samples that match the toy distribution will

Algorithm 1 Euler-Maruyama Predictor on $\mathrm{SO}(3)$

Require: SDE on $\mathrm{SO}(3)$ `SO3SDE`, score network `ScoreNet`, number of steps N_{steps}, noise weight $\lambda(t)$
Ensure: Sample $\mathbf{R}_0$

$\{t_i\}_{i=0}^{N_{\text{steps}}} \leftarrow \text{linspace}(1, 0, N_{\text{steps}} + 1)$
$\Delta t \leftarrow \frac{1}{N_{\text{steps}}}$
$\mathbf{R}_1 \sim \mathcal{U}(\mathrm{SO}(3))$
$\mathbf{R} \leftarrow \mathbf{R}_1$
for $i = 0$ **to** $N_{\text{steps}} - 1$ **do**
  $t \leftarrow t_i$
  $s_\theta(\mathbf{R}, t) \leftarrow \texttt{ScoreNet}(\mathbf{R}, t) \cdot \frac{1}{\lambda(t)}$
  $(b(\mathbf{R}, t), g(t)) \leftarrow \texttt{SO3SDE}(\mathbf{R}, t)$
  $b(\mathbf{R}, t) \leftarrow b(\mathbf{R}, t) - g(t)^2 s_\theta(\mathbf{R}, t)$
  $z \sim \mathcal{N}(0, \mathbf{I})$
  $\mathbf{R} \leftarrow \mathbf{R}\,\mathrm{Exp}\left(b(\mathbf{R}, t)\Delta t\right)$
  $\mathbf{R} \leftarrow \mathbf{R}\,\mathrm{Exp}\left(g(t) z \sqrt{\Delta t}\right)$
end for
$\mathbf{R}_0 \leftarrow \mathbf{R}$
return $\mathbf{R}_0$

build confidence in applying similar $\mathrm{SE}(3)$ equivariant diffusion principles to protein models.

Milestone outcome: a Jupyter notebook demonstrating correct $\mathrm{SO}(3)$ diffusion sampling, which we can later integrate into the BioEmu framework.

4. Stage 2: Fine-tuning BioEmu Demonstration on a Single Protein

Next, we will apply the new fine-tuning method on BioEmu for a single protein sequence to demonstrate a proof-of-concept in a real-world scenario. Prior to this, we will need to fine-tune the toy model of two $\mathrm{IGSO}(3)$ distributions to adjust their mixture weights. Suppose we introduce an additional drift term into the original reverse diffusion process:

$$\begin{aligned} \mathrm{d}\tilde{\mathbf{X}}_t = &\Big(b(\tilde{\mathbf{X}}_t, t) - g(t)^2 \nabla_{\tilde{\mathbf{X}}_t} \log p_t(\mathbf{X}_t)\Big)\mathrm{d}t \\ &+ g(t) u(\tilde{\mathbf{X}}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t^{\mathcal{M}} \end{aligned} \tag{9}$$

and minimize the Kullback-Leibler divergence between the original and perturbed distributions (Hsu, 2002):

$$\begin{aligned} D_{\mathrm{KL}}(\tilde{\mathbf{X}}_0 \parallel \mathbf{X}_0) &\le D_{\mathrm{KL}}(\tilde{\mathbb{P}} \parallel \mathbb{P}) \\ &= \frac{1}{2}\mathbb{E}_{\tilde{\mathbb{P}}}\left[\int_0^1 \left\| u(\tilde{\mathbf{X}}_t, t)\right\|_{\mathcal{M}}^2 \mathrm{d}t\right] \end{aligned} \tag{10}$$

to solve the following constrained optimization problem:

$$\begin{aligned} &\arg\min_u D_{\mathrm{KL}}(\tilde{\mathbf{X}}_0 \parallel \mathbf{X}_0) \\ &\text{s.t.} \quad \mathbb{E}_{\tilde{\mathbf{X}}_0}[h_i(\tilde{\mathbf{X}}_0)] = h_i^*, \quad i = 1, \ldots, N\,. \end{aligned} \tag{11}$$

Then we will select an example protein with an extreme stability phenotype. One candidate is an IDP from the CALVADOS dataset used in the BioEmu paper. Using this protein, our method will enforce this via the constrained optimization above. After fine-tuning on this single sequence, we expect that an IDP's generated conformations will be mostly unfolded, matching experimental observations.

Milestone outcome: a demonstration that our fine-tuning method works on the toy model and can successfully alter the BioEmu's predictions for a specific protein in a manner consistent with experimental stability data.

5. Stage 3: Scaling to MEGAScale Dataset and MD Data

Having validated the approach on a single protein, we will extend it to a large-scale fine-tuning using the MEGAScale dataset of protein folding stabilities, which was also used in training the original BioEmu. For each protein or mutant in the training set, the model will generate an ensemble and compute an expected stability-related quantity. We will then compute the error between these model predictions and the experimental values, and update the model parameters to reduce this error. In addition to the experimental data, we will also attempt to incorporate the molecular dynamics (MD) simulation dataset to provide direct structural physics signals.

Milestone outcome: a fine-tuned version of the BioEmu model that integrates experimental stability data via our SE(3) equivariant constrained fine-tuning method. This model should have hopefully improved accuracy (in terms of predicted vs experimental $\Delta\Delta G$) while retaining physically plausible conformational sampling.

6. Stage 4: Benchmarking and Evaluation

The final stage focuses on rigorous evaluation of the fine-tuned model against benchmarks, and comparison to the original PPFT-based BioEmu. We will use the same evaluation protocols and datasets as the original BioEmu study to ensure a direct comparison. First, on the held-out stability dataset, we will assess the predictive $\Delta\Delta G$ accuracy of our model. Next, we will evaluate if our model has preserved the pretrained distribution aside from the intended shifts in stability-related aspects. For example, we will apply our fine-tuned model to sample ensembles for proteins with known conformational changes or binding events to verify that it still generates diverse, biologically relevant conformations. Finally, we will compare our model's predictions to other computational stability predictors such as single-point mutations with known experimental $\Delta\Delta G$ (ProTherm or the SKEMPI database) and see how well our model's predicted stability change correlates with experiments.

Milestone outcome: a comprehensive benchmark report. We expect to show that our SE(3) equivariant fine-tuning method achieves at least comparable accuracy to PPFT on stability prediction, and we will highlight any improvements.

References

Bortoli, V. D., Mathieu, E., Hutchinson, M. J., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=oDRQGo8I7P.

Hsu, E. P. *Stochastic Analysis on Manifolds*. Number 38. American Mathematical Soc., 2002.

Leach, A., Schmon, S. M., Degiacomi, M. T., and Willcocks, C. G. Denoising diffusion probabilistic models on SO(3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. URL https://openreview.net/forum?id=BY88eBbkpe5.

Lewis, S., Hempel, T., Jiménez-Luna, J., Gastegger, M., Xie, Y., Foong, A. Y. K., Satorras, V. G., Abdin, O., Veeling, B. S., Zaporozhets, I., Chen, Y., Yang, S., Schneuing, A., Nigam, J., Barbero, F., Stimper, V., Campbell, A., Yim, J., Lienen, M., Shi, Y., Zheng, S., Schulz, H., Munir, U., Tomioka, R., Clementi, C., and Noé, F. Scalable emulation of protein equilibrium ensembles with generative deep learning. *bioRxiv*, 2025. doi: 10.1101/2024.12.05.626885. URL https://www.biorxiv.org/content/early/2025/02/25/2024.12.05.626885.

Solà, J., Deray, J., and Atchuthan, D. A micro lie theory for state estimation in robotics, 2021. URL https://arxiv.org/abs/1812.01537.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

A. Brownian Motion on Lie Groups

Proposition A.1 (Marginal Distribution of IGSO(3))**.** *Let* $\mathbf{R}_t \sim \mathrm{IGSO}(3)(\mathbf{R}_0, \sigma^2)$ *be a random rotation matrix. The marginal distribution of its rotation angle* $\omega_t = \|\mathrm{Log}(\mathbf{R}_t)\|_2$ *is given by its PDF* $\frac{1-\cos\omega_t}{\pi} f(\omega_t; \mathbf{R}_0, \sigma)$*, where* $f(\omega_t; \mathbf{R}_0, \sigma)$ *is defined as*

$$f(\omega_t; \mathbf{R}_0, \sigma) = \sum_{\ell=0}^{\infty} e^{-\frac{\sigma^2}{2}\ell(\ell+1)} \frac{\sin\left(\ell+\frac{1}{2}\right)\omega_0}{\sin\frac{\omega_0}{2}} \frac{\sin\left(\ell+\frac{1}{2}\right)\omega_t}{\sin\frac{\omega_t}{2}} \tag{12}$$

Here $\omega_0 = \|\mathrm{Log}(\mathbf{R}_0)\|_2$ *is the rotation angle of* $\mathbf{R}_0$*.*

Proof. The proof follows from the fact that the marginal distribution of ω_t is given by integrating the joint density against the Haar measure $\mathrm{d}\mu_{\mathrm{SO}(3)} = 4\sin^2\frac{\omega}{2}\mathrm{d}\omega \wedge \mathrm{d}\Omega$, constrained to rotations of fixed angle on $\mathbb{S}^2$:

$$\begin{aligned}\frac{1-\cos\omega_t}{\pi} f(\omega_t; \mathbf{R}_0, \sigma) &= 4\sin^2\frac{\omega_t}{2}\int_{\mathbb{S}^2} p(\mathbf{R}_t; \mathbf{R}_0, \sigma)\mathrm{d}\Omega \\ &= \frac{1-\cos\omega_t}{4\pi^2}\sum_{\ell=0}^{\infty}(2\ell+1)e^{-\frac{\sigma^2}{2}\ell(\ell+1)}\int_{\mathbb{S}^2}\chi_\ell(\mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t)\mathrm{d}\Omega\end{aligned} \tag{13}$$

where $\chi_\ell(\mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t)$ denotes the ℓ-th irreducible unitary representation of $2\ell+1$ dimension. Writing the character in terms of Wigner D-matrices:

$$\begin{aligned}\chi_\ell(\mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t) &= \sum_{m=-\ell}^{\ell} D^{(\ell)}_{mm}(\mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t) \\ &= \sum_{m=-\ell}^{\ell}\sum_{n=-\ell}^{\ell} D^{(\ell)}_{mn}(\mathbf{R}_0) D^{(\ell)}_{nm}(\mathbf{R}_t)\end{aligned} \tag{14}$$

Now consider integrating $D^{(\ell)}(\mathbf{R}_t)$ on $\mathbb{S}^2$. For any $\mathbf{R} \in \mathrm{SO}(3)$, the integral over the class of rotations sharing a given rotation angle is invariant under conjugation, which implies

$$\begin{aligned}\int_{\mathbb{S}^2} D^{(\ell)}(\mathbf{R}_t)\mathrm{d}\Omega &= \int_{\mathbb{S}^2} D^{(\ell)}(\mathbf{R}\mathbf{R}_t\mathbf{R}^{-1})\mathrm{d}\Omega \\ &= D^{(\ell)}(\mathbf{R})\left(\int_{\mathbb{S}^2} D^{(\ell)}(\mathbf{R}_t)\mathrm{d}\Omega\right) D^{(\ell)}(\mathbf{R})^{-1}\end{aligned} \tag{15}$$

According to Schur's lemma, this integral must be proportional to the identity matrix, so we can write

$$\begin{aligned}\int_{\mathbb{S}^2} D^{(\ell)}(\mathbf{R}_t)\mathrm{d}\Omega &= \frac{1}{2\ell+1}\,\mathrm{tr}\left(\int_{\mathbb{S}^2} D^{(\ell)}(\mathbf{R}_t)\mathrm{d}\Omega\right)\mathbf{I} \\ &= \frac{4\pi}{2\ell+1}\chi_\ell(\mathbf{R}_t)\mathbf{I}\end{aligned} \tag{16}$$

where $\mathbf{I}$ is the identity matrix. Thus, we can express the integral of $\chi_\ell(\mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t)$ as

$$\begin{aligned}\int_{\mathbb{S}^2}\chi_\ell(\mathbf{R}_0^{\mathrm{T}}\mathbf{R}_t)\mathrm{d}\Omega &= \sum_{m=-\ell}^{\ell}\sum_{n=-\ell}^{\ell} D^{(\ell)}_{mn}(\mathbf{R}_0)\frac{4\pi}{2\ell+1}\chi_\ell(\mathbf{R}_t)\delta_{mn} \\ &= \frac{4\pi}{2\ell+1}\chi_\ell(\mathbf{R}_t)\sum_{m=-\ell}^{\ell} D^{(\ell)}_{mm}(\mathbf{R}_0) \\ &= \frac{4\pi}{2\ell+1}\chi_\ell(\mathbf{R}_0)\chi_\ell(\mathbf{R}_t)\end{aligned} \tag{17}$$

Substituting this back into the marginal distribution finally gives

$$\begin{aligned} f(\omega_t; \mathbf{R}_0, \sigma) &= \frac{1}{4\pi} \sum_{\ell=0}^{\infty} (2\ell+1) e^{-\frac{\sigma^2}{2}\ell(\ell+1)} \int_{\mathbb{S}^2} \chi_\ell(\mathbf{R}_0^{\mathrm{T}} \mathbf{R}_t) \mathrm{d}\Omega \\ &= \sum_{\ell=0}^{\infty} e^{-\frac{\sigma^2}{2}\ell(\ell+1)} \frac{\sin\left(\ell+\frac{1}{2}\right)\omega_0}{\sin\frac{\omega_0}{2}} \frac{\sin\left(\ell+\frac{1}{2}\right)\omega_t}{\sin\frac{\omega_t}{2}} \end{aligned} \tag{18}$$

□

B. Fine-tuning diffusion models on Riemannian manifolds

$$\begin{aligned} \mathcal{L}(\theta) &= \mathbb{E}_{\tilde{\mathbb{P}}_\theta}[L_\theta(\mathbf{X})] \\ &= \mathbb{E}_{\tilde{\mathbb{P}}_{\mathrm{sg}(\theta)}}[w_\theta(\mathbf{X}) L_\theta(\mathbf{X})] \end{aligned} \tag{19}$$

$$\begin{aligned} w_\theta(\mathbf{X}) &= \frac{\mathrm{d}\tilde{\mathbb{P}}_\theta}{\mathrm{d}\tilde{\mathbb{P}}_{\mathrm{sg}(\theta)}}(\mathbf{X}) \\ &= \exp\left(\int_0^1 \left\langle u_\theta(\mathbf{X}_t, t) - u_{\mathrm{sg}(\theta)}(\mathbf{X}_t, t), \mathrm{d}\mathbf{W}_t^{\mathcal{M}} \right\rangle_{\mathcal{M}} - \frac{1}{2} \int_0^1 \left\| u_\theta(\mathbf{X}_t, t) - u_{\mathrm{sg}(\theta)}(\mathbf{X}_t, t) \right\|_{\mathcal{M}}^2 \mathrm{d}t \right) \end{aligned} \tag{20}$$

$$\begin{aligned} \frac{\partial}{\partial\theta}\mathcal{L}(\theta) &= \mathbb{E}_{\tilde{\mathbb{P}}_{\mathrm{sg}(\theta)}}\left[\frac{\partial}{\partial\theta}(w_\theta(\mathbf{X}) L_\theta(\mathbf{X})) \right] \\ &= \mathbb{E}_{\tilde{\mathbb{P}}_{\mathrm{sg}(\theta)}}\left[w_\theta(\mathbf{X}) \frac{\partial}{\partial\theta} L_\theta(\mathbf{X}) \right] + \mathbb{E}_{\tilde{\mathbb{P}}_{\mathrm{sg}(\theta)}}\left[\frac{\partial}{\partial\theta} w_\theta(\mathbf{X}) L_\theta(\mathbf{X}) \right] \end{aligned} \tag{21}$$

$$\begin{aligned} \frac{\partial}{\partial\theta} w_\theta(\mathbf{X}) &= w_\theta(\mathbf{X}) \int_0^1 \left\langle \nabla_\theta u_\theta(\mathbf{X}_t, t), \mathrm{d}\mathbf{W}_t^{\mathcal{M}} \right\rangle_{\mathcal{M}} \\ &= \int_0^1 \left\langle \nabla_\theta u_\theta(\mathbf{X}_t, t), \mathrm{d}\mathbf{W}_t^{\mathcal{M}} \right\rangle_{\mathcal{M}} \end{aligned} \tag{22}$$

$$\hat{\mathcal{L}}_{\mathrm{EV}}(\theta) = \sum_{i=1}^{N} \left(\left(\frac{1}{M} \sum_{j=1}^{M} w_\theta(\mathbf{X}^{(j)}) h_i(\mathbf{X}_0^{(j)}) - h_i^* \right)^2 - \frac{1}{M(M-1)} \sum_{j=1}^{M} \left(w_\theta(\mathbf{X}^{(j)}) h_i(\mathbf{X}_0^{(j)}) - \frac{1}{M} \sum_{j=1}^{M} w_\theta(\mathbf{X}_0^{(j)}) h_i(\mathbf{X}_0^{(j)}) \right)^2 \right) \tag{23}$$

$$\hat{\mathcal{L}}_{\mathrm{KL}}(\theta) = \frac{1}{2M} \sum_{j=1}^{M} \left(w_\theta(\mathbf{X}^{(j)}) \int_0^1 \left\| u_\theta(\mathbf{X}_t^{(j)}, t) \right\|_{\mathcal{M}}^2 \mathrm{d}t \right) \tag{24}$$

$$\hat{\mathcal{L}}(\theta) = \hat{\mathcal{L}}_{\mathrm{EV}}(\theta) + \lambda \hat{\mathcal{L}}_{\mathrm{KL}}(\theta) \tag{25}$$