DENOISING DIFFUSION PROBABILISTIC MODELS ON SO(3) FOR ROTATIONAL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Probabilistic diffusion models are capable of modeling complex data distributions on high-dimensional Euclidean spaces for a range applications. However, many real world tasks involve more complex structures such as data distributions defined on manifolds which cannot be easily represented by diffusions on $\mathbb{R}^n$. This paper proposes denoising diffusion models for tasks involving 3D rotations leveraging diffusion processes on the Lie group $SO(3)$ in order to generate candidate solutions to rotational alignment tasks. The experimental results show the proposed $SO(3)$ diffusion process outperforms naïve approaches such as Euler angle diffusion in synthetic rotational distribution sampling and in a 3D object alignment task.

1 INTRODUCTION

Denoising diffusion probabilistic models are capable of generating high quality samples from complex distributions and have delivered encouraging results in audio synthesis and image applications. However, there are many problems such as pose estimation and protein docking for which the domain $\mathbb{R}^n$ is unsuitable. As many of these problems are roto-translational in nature, sampling rotations from a conditional diffusion model allows for a probabilistic model of possible poses. In this work, we introduce denoising diffusion models on the Lie group of 3D rotations, $SO(3)$.

Denoising diffusion probabilistic models (DDPMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020) are a set of generative models inspired by non-equilibrium thermodynamics. The underlying idea consists of simulating a diffusion process that takes some form of observed data (e.g. images), denoted $\mathbf{x}_0$, with unknown distribution $q(\mathbf{x}_0)$ and transforms (diffuses) it into pure noise. A generative model can thus be found by learning the reverse process, turning noise back into the structure of the underlying data.

In practice, the diffusion is replaced by a non-homogenous discrete time Markov chain with one-step transition density. The distribution at step t of the forward process Markov chain $q(\mathbf{x}_t)$ is conditional on only the step before it, $q(\mathbf{x}_t|\mathbf{x}_{t-1})$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}), \tag{1}$$

where $\beta_t, t = 1, \ldots, T$ denotes a variance schedule and $\mathcal{N}(y; \mu, \Sigma)$ a Gaussian density with argument y, mean μ and covariance matrix Σ. Under appropriate conditions, the final value $\mathbf{x}_T$ will approximately follow an Gaussian distribution $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$.

Denoising models learn an approximation of the reverse process $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ where $p(\mathbf{x}_T) = \mathcal{N}(0, \mathbf{I})$. The transition kernel $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ hence learns to predict the previous time step of the forward process and is parameterized by a normal distribution

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)). \tag{2}$$

The functions μ_θ and Σ_θ are implemented as outputs of a neural network with learned parameters θ. More recent work (Ho et al., 2020) suggests that taking the covariance matrix Σ_t in equation 2 fixed can result in better performance. Further reparameterization of the forward process $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + (1-\bar{\alpha}_t)\epsilon$. where $\epsilon \sim \mathcal{N}(0, I)$ results in

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right). \tag{3}$$

The loss equation then can be simplified to a function of added noise.

$$L_t(\theta) = \mathbb{E}_{\tau,\epsilon,\mathbf{x}_0}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_\tau, \tau)\|^2\right]. \tag{4}$$

2 Defining a diffusion on the rotation group SO(3)

A key component of training diffusion models is the ability to sample from the diffusion distribution at time t without having to calculate intermediate values. For the normal distribution in $\mathbb{R}^n$ (Euclidean diffusion), this can be accomplished easily through the previously derived closed form equations. However, this is not easily generalised to the space of rotations $SO(3)$ due to several factors.

A naïve approach to diffusion on $SO(3)$ would be to use Euler angles, treating the rotations (ψ, ϑ, ϕ) as diffusion over an $\mathbb{R}^3$ space. However, the nature of diffusions over Euler angles means that they cannot correctly capture symmetries present in rotational systems. Alternatively, rotations may be represented as unit quaternions. However, a diffusion process treating the quaternion representation as $\mathbb{R}^4$ would require mapping the diffused position to the unit sphere at every step. The fast sampling scheme previously defined (Ho et al., 2020) would not be usable due to this nonlinear mapping. Similarly, a Gaussian distribution on $\mathbb{R}^4$ conditioned to lie on the 3-sphere is described by the Bingham distribution (Bingham, 1974). Additionally, while quaternions are a continuous representation of rotations, they form a double cover of $SO(3)$. This results in each rotation being represented by two equally valid quaternions, causing difficulties for neural networks (Zhou et al., 2019).

Instead of attempting to use an $\mathbb{R}^n$ normal distribution on the space of rotations, we instead look for an analogous distribution defined on $SO(3)$. Before we proceed we outline some of the desirable properties of Gaussian distributions that are required for the distribution on $SO(3)$. For example, if $p_X(x), p_Y(y)$ are (independent) normal distributions defined as $\mathcal{N}(\mu_x, \sigma_x^2), \mathcal{N}(\mu_y, \sigma_y^2)$ respectively, then the distribution of their sum $p_Z(z)$ can be written as $Z \sim N(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$. For this reason, the normal distribution is said to be closed under convolution as the convolution of two normal densities (corresponding to the density of the sum) is also a normal distribution. This property is key in allowing us to derive the equation for the distribution of $q(\mathbf{x}_t|\mathbf{x}_0)$ and sample a diffusion model efficiently. A distribution for a diffusion on $SO(3)$ must also have this property in order to derive equations for efficient sampling of the forward process.

The Fisher matrix and Bingham distributions have previously been used in deep learning as probabilistic rotation estimators (Mohlin et al., 2020; Gilitschenski et al., 2020) for ortho-normal matrix and quaternion representations of SO(3) respectively. However, neither of these distributions are closed under convolution (Schaeben & Nikolayev, 1998; Glover & Kaelbling, 2013), making quantifying the rotational distribution at arbitrary timesteps difficult.

Instead, we consider the isotropic Gaussian distribution on SO(3) (IG) (Savjolova, 1985) $g \sim \mathcal{IG}_{SO(3)}(\mu, \epsilon^2)$, parameterized by a mean rotation μ and scalar variance ϵ. The IG distribution can be parameterized in an axis-angle form, with uniformly sampled axes and rotation angle $\omega \in [0, \pi]$ with density

$$f(\omega) = \frac{1 - \cos\omega}{\pi} \sum_{l=0}^{\infty} (2l+1) e^{-l(l+1)\epsilon^2} \frac{\sin((l+\frac{1}{2})\omega)}{\sin(\omega/2)}. \tag{5}$$

Note that the uniform distribution on SO(3), denoted $\mathcal{U}_{SO(3)}$, is parameterized with uniform-axis and $f(\omega) = \frac{1-\cos\omega}{\pi}$ and needs to be included as a scaling factor when sampling from the distribution.

The IG distribution is both a natural extension of the central limit theorem (CLT) and the expected distribution of Brownian motion (Qiu, 2013) on $SO(3)$, providing a strong motivation to use it for denoising diffusion models. While the variance of the IG distribution is defined as a scalar value, and thus has less flexibility than the Matrix-Fisher or Bingham distributions, Euclidean denoising diffusion models assume no correlation between dimensions, and are thus also parameterized with a scalar variance. Most importantly, the IG distribution is closed under convolution.

We can use this relationship to derive similar equations to the Euclidean diffusion process, allowing us to define the distribution at arbitrary timesteps for efficient sampling. Initially, data $\mathbf{x}_0 \in SO(3)$ is sampled from the distribution $q(\mathbf{x}_0)$ and diffused. As before, at $t = T$ we consider the data to be fully diffused.

To derive the distribution $q(\mathbf{x}_t|\mathbf{x}_0)$, note that the analogous Euclidean distribution (Equation 1) requires a scaling term to be applied to the $\mathbf{x}_0$ term. A scaling term applied directly to a rotation matrix is nonsensical, as the resulting value does not lie on the $SO(3)$ manifold. If $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ is viewed as a translation away from the origin, it implies that $\mathbf{x}_0 \sim s(\mathbf{x}_0)$ can be viewed as a rotation away from the identity rotation $\mathbf{I}$, and we can scale our rotations by interpolating the angle of rotation along the geodesic from the identity. For this, we rely upon the exponential and logarithmic maps between the Lie algebra $\mathfrak{so}(3)$ and $SO(3)$. Intuitively, a rotation matrix R has an associated angle of rotation θ, and a rotation matrix $P = RR = R^2$ has an angle of rotation of 2θ. We follow standard definitions (Cardoso & Leite, 2010) of the logarithm of a rotation matrix and define it as

$$\log R = \frac{\theta}{2\sin\theta}\left(R^\top - R\right)$$

where θ satisfies $1 + 2\cos\theta = \text{trace}(R)$. Matrices in $\mathfrak{so}(3)$ are skew-symmetric of form $S(v)$ with

$$S(v) = \begin{pmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{pmatrix}, \qquad v = [x, y, z]\,,$$

where $\|v\|_2 = \theta$. From this definition of the rotation matrix logarithm, we are able to scale rotation matrices by converting them to values in the Lie algebra $\mathfrak{so}(3)$, element-wise multiplying by a scalar value and converting back to rotation matrices through matrix exponentiation. The composition of rotations is done through matrix multiplication in $SO(3)$, analogous to addition in Euclidean diffusion models:

$$\lambda(\gamma, \mathbf{x}) = \exp\left(\gamma \log\left(\mathbf{x}\right)\right). \tag{6}$$

Applying these to equations from the original DDPM model we arrive at the following definitions:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{IG}_{SO(3)}(\lambda(\sqrt{\bar{\alpha}_t}, \mathbf{x}_0)), (1 - \bar{\alpha}_t)); \tag{7}$$

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{IG}_{SO(3)}(\tilde{\mu}(\mathbf{x}_t\mathbf{x}_0), \tilde{\beta}_t) \tag{8}$$

and

$$\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0) = \lambda\left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}, \mathbf{x}_0\right)\lambda\left(\frac{\sqrt{\alpha_{t-1}}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}, \mathbf{x}_t\right). \tag{9}$$

3 Experiments

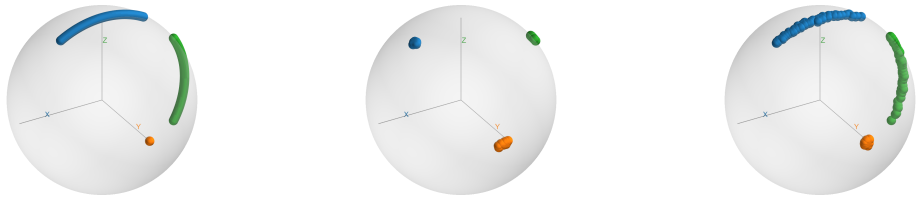
3.1 Learning Data Distributions on SO(3)

To evaluate the validity of the proposed SO(3) diffusion process, a synthetic data distribution was created. This distribution consists of rotations about the z-axis of $\pm 90°$, with equal chance of either rotation. A fully connected feed-forward network was trained using Algorithm 1. Inputs to the network consisted of raw rotation matrices and sine-cosine encoded t values.

Rotations generated from the reverse diffusion process closely approximate the initial data distribution. After generating 512 samples from the reverse process, the mean angular error between a sample and the closest rotation in the synthetic dataset was 0.0239 radians. The 512 samples were then categorised into z_{+90} (261 successes) and z_{-90} groups and modeled as a binomal distribution with probability of success p. Using a Beta$(\alpha = 1, \beta = 1)$ distribution as the conjugate prior, $\mathbb{P}(0.45 < p < 0.55 \mid z_{1:512}) = 0.963$, showing that the reverse process has also modeled the split of the data correctly.

While this shows that the denoising diffusion on $SO(3)$ is capable of learning a data distribution, it does not show whether it is any more capable than Euclidean diffusion. To test this, we consider an idiomatic approach to diffusion of rotations: Euclidean diffusion of Euler angles; and show that this approach breaks down when the data distribution passes through singularities in the Euler representation.

We construct a data distribution of uniformly sampled rotations about the y-axis between $\frac{\pi}{3}$ and $\frac{2\pi}{3}$ radians (Figure 2a) and train two diffusion processes with equal network capacity until convergence. Figure 2b shows samples from the trained Euclidean–Euler process, revealing an inability to correctly capture the distribution, whereas the $SO(3)$ diffusion process (Figure 2c) provides a significantly better way of approximating the distribution and sampling.



(a) Target Distribution (b) Euler Angle Diffusion (c) $\mathcal{IG}_{SO(3)}$ Diffusion

Figure 1: Visualisation of the columns of sampled rotation matrices, corresponding to the transformation of basis vectors. Euler angle diffusion is unable to reconstruct the distribution correctly due to passing through the singularity at $\theta = \frac{\pi}{2}$.

Distribution	$\widehat{\text{MMD}}_b$	$p < 0.05$
Small Uncorrelated Rotations	0.0433	Yes
Large Uncorrelated Rotations	0.0027	Yes
Large Correlated Rotations	0.0317	Yes

Table 1: MMD estimates between generated samples and samples taken from Quaternion-Bingham distributions (See Appendix D). Significance testing shows that the diffusion model correctly captures various distributions.

To quantify the differences between the target distribution and samples generated from the diffusion process, we repeat this experiment with Quaternions sampled from Bingham distributions over $\mathcal{S}^3$, using the maximum mean discrepancy (Gretton et al., 2012), with $k(x, y) = \exp(-|| \log(x^{-1}y)||_F)$, a kernel over the $SO(3)$ geodesic distance.

3.2 Rotational Alignment with SO(3) Diffusion Processes

By using an $SO(3)$ diffusion process to represent the rotation of a pointcloud, we can use denoising diffusion models to generate solutions for a rotational alignment problem. In order to test this, an alignment problem using point-clouds of aircraft from ShapeNet (Chang et al., 2015) was designed. This dataset consists of pre-aligned point-clouds of various real and fictional aircraft. As the shape and style of these aircraft differ greatly, the task of rotating them to the correct orientation, in effect detecting the forward direction of the aircraft, relies on being able to generalize over common features. At each training iteration, rotations at arbitrary timesteps were generated and used to rotate samples from the dataset. The network architecture uses a non-causal transformer, with each token consisting of SIREN (Sitzmann et al., 2020) encoded coordinates concatenated with a sinusoidal positional embedding of the timestep. We compare angular errors from Euler and $\mathcal{IG}_SO(3)$ diffusion models.

	Percentile					
Method	1%	5%	10%	50%	90%	95%
Euler	0.64	1.09	1.37	2.26	2.97	3.05
$\mathcal{IG}_{SO(3)}$	**0.01**	**0.02**	**0.03**	**0.16**	**2.94**	**3.03**

Table 2: Distribution of angular error (radians) in an aircraft alignment task. Our method outperforms the Euler angle diffusion method across all percentiles.

4 Conclusion

In conclusion, we have introduced a denoising diffusion probabilistic model over $SO(3)$ and shown that it is capable of correctly learning distributions over the space of rotations. Furthermore, we used this model as a framework for a synthetic alignment task, showing significantly lower errors than a naïve approach. This paper acts as an introduction to probabilistic sampling methods for alignment tasks. In the future, we aim to extend this method to $SE(3)$ and perform roto-translational alignment. In particular, we believe diffusion models are a suitable model for protein receptor-ligand interactions and can be used to probabilistically model the solution space to protein docking problems.

REFERENCES

Christopher Bingham. An antipodally symmetric distribution on the sphere. *Annals of Statistics*, 2: 1201–1225, 1974.

João R. Cardoso and F. Silva Leite. Exponentials of skew-symmetric matrices and logarithms of orthogonal matrices. 233(11):2867–2875, 2010. ISSN 0377-0427. doi: 10.1016/j.cam.2009.11.032. URL https://www.sciencedirect.com/science/article/pii/S0377042709007791.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3d model repository. 2015. URL http://arxiv.org/abs/1512.03012.

Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep orientation uncertainty learning based on a bingham loss. 2020. URL https://iclr.cc/virtual_2020/poster_ryloogSKDS.html.

Jared Glover and Leslie Pack Kaelbling. Tracking 3-d rotations with the quaternion bingham filter. 2013. URL https://dspace.mit.edu/handle/1721.1/78248. Accepted: 2013-03-29T21:30:06Z.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 33: 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

S. Matthies, J. Muller, and G. W. Vinel. On the normal distribution in the orientation space. *Textures and Microstructures*, 10(1):77–96, January 1988. doi: 10.1155/tsm.10.77. URL https://doi.org/10.1155/tsm.10.77.

David Mohlin, Josephine Sullivan, and Gérald Bianchi. Probabilistic orientation estimation with matrix fisher distributions. 33:4884–4893, 2020. URL https://papers.nips.cc/paper/2020/hash/33cc2b872dfe481abef0f61af181dfcf-Abstract.html.

Dmitry I. Nikolayev and Tatjana I. Savyolov. Normal distribution on the rotation group so(3). 29(3): 201–233, 1997. ISSN 1687-5397. doi: 10.1155/TSM.29.201. URL https://www.hindawi.com/journals/tsm/1997/173236/. Publisher: Hindawi.

Yu Qiu. Isotropic distributions for 3-dimension rotations and one-sample bayes inference. 2013. doi: https://doi.org/10.31274/etd-180810-3598. URL https://lib.dr.iastate.edu/etd/13007.

T. I. Savjolova. *Preface to* novye metody issledovanija tekstury polikristalliceskich materialov. *Metallurgija, Moscow*, 1985.

Helmut Schaeben and Dmitry I. Nikolayev. The central limit theorem in texture component fit methods. 53(1):59–87, 1998. ISSN 1572-9036. doi: 10.1023/A:1005979525038. URL https://doi.org/10.1023/A:1005979525038.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015. URL http://proceedings.mlr.press/v37/sohl-dickstein15.html. ISSN: 1938-7228.

Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

A PARAMETERIZATION OF REVERSE PROCESS

When training or sampling from a euclidean diffusion model, the stochastic part of the forward and reverse processes acan be calculated by scaling a standard normal distribution. In DDPMs (Ho et al., 2020), the reverse process mean is given by Equation 3: As DDPMs resemble Langevin dynamics, with ϵ_θ predicting a learned gradient of the data density, we run into the issue of what to predict in our $SO(3)$ model. We cannot directly predict values in $SO(3)$ as the gradient of a rotation matrix is not a rotation matrix. Instead, the gradient of a rotation matrix R lies on the tangent space $T_R SO(3)$ and takes the form $S(v)R$, where $S(v)$ is a skew symmetric matrix. Predicting the value $S(v)R$ directly involves predicting a point lying on a 3D hyperplane in a 9D space, and requires network knowledge of the rotation $R = \mathbf{x}_t$. We simplify the job of the network by predicting v instead, noting that $S(v)$ can also be interpreted as an angular velocity tensor. Secondly, sampling from $\mathcal{IG}_{SO(3)}(I, \lambda)$ cannot be done by scaling samples taken from $\mathcal{IG}_{SO(3)}(I, 1)$. Instead, during training (Algorithm 1), we sample our diffusion rotation matrix $R \sim \mathcal{IG}_{SO(3)}(\mathbf{I}, \sqrt{1-\bar{\alpha}_t})$, convert into skew-symmetric form through taking the matrix-logarithm and scale the target by $\frac{1}{\sqrt{1-\bar{\alpha}_t}}$ in order to have a target analogous to the standard normal distribution used when training an euclidean diffusion model. This parameterization allows us to follow a similar scheme to euclidean diffusion models when sampling (Appendix C)

B SAMPLING FROM THE ISOTROPIC GAUSSIAN ON SO(3) DISTRIBUTION

First we note that a rotation sampled from $\mathcal{IG}_{SO(3)}(\mu, \epsilon^2)$ can be decomposed by sampling a pair of rotations from $(\mathcal{IG}_{SO(3)}(\mu, 0) = \mu,\ \mathcal{IG}_{SO(3)}(I, \epsilon^2))$ where I is the identity rotation. This corresponds to rotating samples taken from an identity-mean distribution by a constant μ. We further decompose $\mathcal{IG}_{SO(3)}(I, \epsilon^2)$ into an axis-angle form. The axis of rotation is uniformly distributed over $\mathcal{S}^2$, allowing for easy sampling, whereas the PDF of the angle of rotation $f(\omega) : \omega \in [0, \pi]$ is given by: (Nikolayev & Savyolov, 1997)

$$f(\omega) = \frac{1-\cos\omega}{\pi} \sum_{l=0}^{\infty} (2l+1) e^{-l(l+1)\epsilon^2} \frac{\sin((l+\frac{1}{2})\omega)}{\sin(\omega/2)} \tag{10}$$

Due to the $e^{-l(l+1)\epsilon^2}$ term in this equation, convergence of this equation is poor for small values of ϵ. We adapt the approximation derived by (Matthies et al., 1988), suitable for $\epsilon \leq 1$.

$$f(\omega) = \frac{(1-\cos(\omega))}{\pi} \sqrt{\pi}\epsilon^{-\frac{3}{2}} e^{\frac{\epsilon}{4}} e^{-\frac{\left(\frac{\omega}{2}\right)^2}{\epsilon}} \cdot \frac{\left[\omega - e^{-\frac{\pi^2}{\epsilon}} \left((\omega - 2\pi) e^{\frac{\pi\omega}{\epsilon}} + (\omega + 2\pi) e^{-\frac{\pi\omega}{\epsilon}}\right)\right]}{2\sin\left(\frac{\omega}{2}\right)} \tag{11}$$

We sample from an approximation of $f(t)$ using an inverse-transform sampling process, a numerical approximation of the CDF is taken through trapezoidal integration of the PDF, with a bias of more samples near 0 as to more accurately capture the distribution for small ϵ. Linear interpolation of uniform samples $[0, 1]$ are then used to to approximate sampling from $f(t)$.

Once an angle has been sampled, an arbitrary axis is chosen from the uniform distribution, and the rotation composed with the mean rotation of the initial distribution. While in general the order of operations matters in SO(3), the isotropic nature of an identity-mean IG distribution means the order of matrix-multiplication does not affect the final distribution. Consider a sample from the distribution $\mathcal{IG}_{SO(3)}(\mu, \epsilon)$, which can be decomposed as $\mu\mathbf{z}_1$ or $\mathbf{z}_2\mu$. As $\mu\mathbf{z}_1 = \mathbf{z}_2\mu$, and $\mu\mathbf{z}_1\mu^-1 = \mathbf{z}_2$, the matrices $\mathbf{z}_1$ and $\mathbf{z}_2$ are similar. Note that this implies $\mathrm{tr}(\mathbf{z}_1) = \mathrm{tr}(\mathbf{z}_2) = 1 + 2\cos\theta$, where θ is the angle of rotation. As $\mathbf{z}_1$ and $\mathbf{z}_2$ share the same angle of rotation and differ only in axis, which is uniformly sampled, the distributions of $\mu\mathbf{z}_1$ and $\mathbf{z}_2\mu$ are the same.

C TRAINING AND SAMPLING ALGORITHMS FOR $SO(3)$ DIFFUSION

The neural network ϵ_θ learns an approximation of the gradient of the data density at each timestep t. In order to train this network we need to sample from $q(\mathbf{x}_t)$ efficiently. Once trained, the network is then used to generate samples matching the distribution $q(\mathbf{x}_0)$.

Algorithm 1 Training

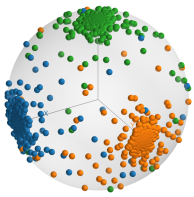
repeat
  $t \sim \text{Uniform}(\{1, ..., T\})$
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $R \sim \mathcal{IG}_{SO(3)}(\mathbf{I}, \sqrt{1-\bar{\alpha}_t})$
  $S(v) = \frac{\log(R)}{\sqrt{1-\bar{\alpha}_t}}$
  $\mathbf{x}_{scale} = \exp\left(\sqrt{\bar{\alpha}}_t \log \mathbf{x}_0\right)$
  Take gradient descent step on:
    $\nabla_\theta ||v - \epsilon_\theta(R\mathbf{x}_{scale}, t)||_2$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{U}_{SO(3)}$
for $t = T, ..., 1$ **do**
  if $t > 1$ **then**
    $R \sim \mathcal{IG}_{SO(3)}(\mathbf{I}, \tilde{\beta}_t)$
  else
    $R = I$
  end if
  $v = \epsilon_\theta(\mathbf{x}_t, t)$
  $\mathbf{a}_1 = \exp(\frac{1}{\sqrt{\bar{\alpha}_t}} \log(\mathbf{x}_t))$
  $\mathbf{a}_2 = \exp(S(\frac{1}{\sqrt{1-\bar{\alpha}_t}} v))$
  $\tilde{\mathbf{x}}_0 = \mathbf{a}_1 \mathbf{a}_2^{-1}$
  $\mathbf{x}_{t-1} = \tilde{\nu}(\mathbf{x}_t, \tilde{\mathbf{x}}_0) R$
end for

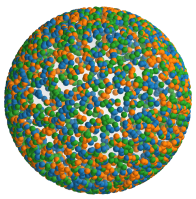
D Visualisation of Quaternion-Bingham Distributions

The distributions chosen in Table 1 were chosen to cover a range of possible distributions. A Bingham distribution is defined over S^n by an $(n+1) \times (n+1)$ covariance matrix. It can be thought of as samples from the Gaussian distribution in $\mathbb{R}^{n+1}$ normalised to unit length. As unit vectors in $\mathbb{R}^4$ can be interpreted as quaternion rotations of the form $(x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k})$, we use this distribution to generate samples in $SO(3)$



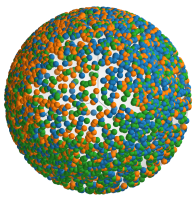
$$\begin{bmatrix} 10^3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) Small Uncorrelated Rotations



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b) Large Uncorrelated Rotations



$$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.9 & 0.9 \\ 0.0 & 0.9 & 1.0 & 0.9 \\ 0.0 & 0.9 & 0.9 & 1.0 \end{bmatrix}$$

(c) Large Correlated Rotations

Figure 2: Visualisation of the basis vectors of 512 samples of the each Quaternion-Bingham distribution and their respective co-variance matrices.