Physics aware joint inference for the cryo-EM inverse problem: normal modes, global 3D pose and CTF defocus.

Geoffrey Woollard
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
gw@cs.ubc.cs

1 Abstract

Inverse imaging problems are common in science, and can be approached with stochastic variational amortized inference. Here we focus on one such inverse problem, single particle electron cryomicroscopy (cryo-EM) of biomolecules, and employ a probabilistic programming perspective where the decoder uses physics equations of the forward model of image formation to map latents to observes (i.e. simulated data, including noise), and the encoder predicts the posterior of the latents from observes (i.e. here I use simulated data to stand in for experimentally measured data). The cryo-EM forward model employed accounts for the effect of the microscope (contrast transfer function), the unknown global 3D pose of the biomolecule, and continuous single particle heterogeneity. I model heterogeneity as an eigendecomposition of an energy based model to second order whose Hessian has a convenient analytical form based on distances between pseudo-atoms. Learning more of the latent space in the forward model from cryo-EM images should allows the conformational ensemble of atomic positions of a biomolecule to be disentangled, which are of basic scientific interest and have applications in domains such as material design and drug development–small molecule and biologics–in the pharmaceutical industry.

2 Introduction

Biomolecular structures are the angstrom-scale formal causes that underlie the unity of a whole living organism. Structural biology represents the flow of energy and information of molecular life in a visual manner. Humans find images and movies of biomolecules intuitive for thinking about causal relationships that mirror physical interactions they are familiar with, and that corresponds to physical modelling that draws on the theories of electrostatics and statistical thermodynamics [6]. Since Perutz and Kendrew's Nobel in 1962, various Nobel prizes have been granted for solving atomic resolution bio-molecular structures, and the Nobel coverage emphasizes the importance for basic biological discovery and applied biomedical application, including the Chemistry Nobel in 2017 "for developing cryo-electron microscopy for the high-resolution structure determination of biomolecules in solution".

While some structural biology techniques are spectroscopic, single particle electron cryomicroscopy (cryo-EM) is an imaging technique for obtaining magnified images of biomolecules. In a typical cryo-EM experiment a physical liquid sample is prepared of a solubilized biomolecule–each biomolecule has $\sim 10^4 - 10^5$ atoms–in a biochemically homogenous/purified aqueous environment. A few microliters are placed on an electron microscope grid, thinned out through wicking, immobilized through cryogenic freezing, spread on a 2D supporting grid, and imaged in a transmission electron microscope [7]. Thousands of images ($\sim 4000^2$ pixels), each with hundreds of copies of the same

Preprint. Under review.

type of biomolecule ("particle", about $100^2 - 500^2$ pixels) are captured at different 3D poses, and data processing algorithms reconstruct the underling atomic structure.

Pharmaceutical companies and academic research institutes have heavily invested in equipment and personnel for cryo-EM over the last $\sim 5-7$ years. After months or even years of biochemical sample optimization, a homogeneous biomolecular preparation is imaged with a high resolution transmission electron microscope, and data processing algorithms reconstruct the underling atomic structure, or distribution of structures.

I propose a method to learn an ensemble of atomic structures directly from raw 2D cryo-EM measurements, performing inference on continuous conformational heterogeneity, the defocus of the contrast transfer function (CTF), and global rotation. I choose to approach this problem in a stochastic variational amortized inference setting, using the deep probabilistic programming framework Pyro [23, 4, 12]. I take a stochastic variational amortized inference approach (similar to [18]) as a point of departure, with the following features:

1. Perform inference on global 3D pose (rotation in $\mathrm{SO}(3)$), conformational heterogeneity, and CTF defocus through deep encoder neural network architectures.
2. Scale the forward model to tens of thousands of pseudo-atoms in a large box size, with an fast approximate projection of the biomolecular potential via a Gaussian kernel applied to pseudo-atoms.
3. Characterize the posterior of the global rotation with a mixture of Projected Normal distributions – a directional distribution similar to the Von-Misses or Kent distributions, but reparametrizable and therefore suitable for training [9].

3 Related Work

Currently, most structural biology research project uses various software packages [19, 17, 22, 15] to transform raw 2D microsocpe images into one or more voxelized 3D maps. The "reconstruction problem" that averages 2D images together into a 3D map of the Coulumbic density historically grew out of a tradition of digital signal processing and computerized tomography [10, 20, 15]. Instead of representing the biomolecular potential as a 3D voxelized arrray, recent work has focused on an atom or pseudo-atom encoding [26, 18].

In Cryofold [26], Zhong and co-authors represent the biomolcule as a set of coarse grained pseudo-atoms. They learn each Gaussian centre, and global intensity and global Gaussian variance, which control how intense and how spread out the Gaussian kernels are. They use a variational auto-encoder (VAE) to learn offsets to Gaussian centers. The loss term includes harmonic terms that restrain (1) the protein (a type of biomolecule) backbone and side chain pseudo-atoms and (2) consecutive backbone pseudo-atoms each around a respective global reference value, which takes into account the polymeric nature of a protein polypeptide. No source code is available.

In atomVAE, Rosenbaum and authors from DeepMind proposed a method that learns a conformational ensemble from synthetic cryo-EM measurements using a variational auto-encoder approach [18]. They used a multilayer perceptrons (MLP) neural network architecture to learn the 3D pose and conformation of an coarse grained atomic representation, where each amino acid residue is one Gaussian sphere. The sampled pseudo-atoms are projected through a simple model of image formation that treats each residue as a spherical Gaussian density. The simulated image is convoluted with the (known/fixed) microscope parameters, which are not learned. All distributions used to sample are Gaussian. The paper references a backbone continuity loss, to regularize the output of the conformational encoder and thus keep it close to the reference conformation. However no details (equations, etc) are given. No source code is available.

4 Method

4.1 General overview

The forward model is a physics aware decoder that maps the latent space of physically interpretable distribution parameters through linking functions that act as a high fidelity stochastic physics simulator,

which incorporates information about the distribution of unknown parameters. The inverse guide function that maps from observable to latent is a series of deep neural net encoders, or a guide for stochastic variational inference in Pyro's `pyro.infer.SVI` [4]. Here the rotation and microscope effects are typically considered nuisance variables for most research questions, and the structural biologist is interested in characterizing the posterior on the conformational heterogeneity. Under the problem formulation chosen, each particle maps to a distribution of atomic states characterized by the posterior of the conformational heterogeneity, which can be sampled from, and this is represented by the distribution of the scalar component of perturbation eigenvector fields from a type of elastic network models called anisotropic network models, and will be referred to as "normal modes". A graphical model is shown in Figure 1 and a schematic of deep neural network architecture in the learned posterior is shown in Figure 2. Here I simply use three neural networks (one for each of the three latents normal mode, pose, CTF) with convolutional neural network (CNN) and MLP based architecture with no conditioning or weight sharing, an encoder architectures similar to those previously published in [16, 13].

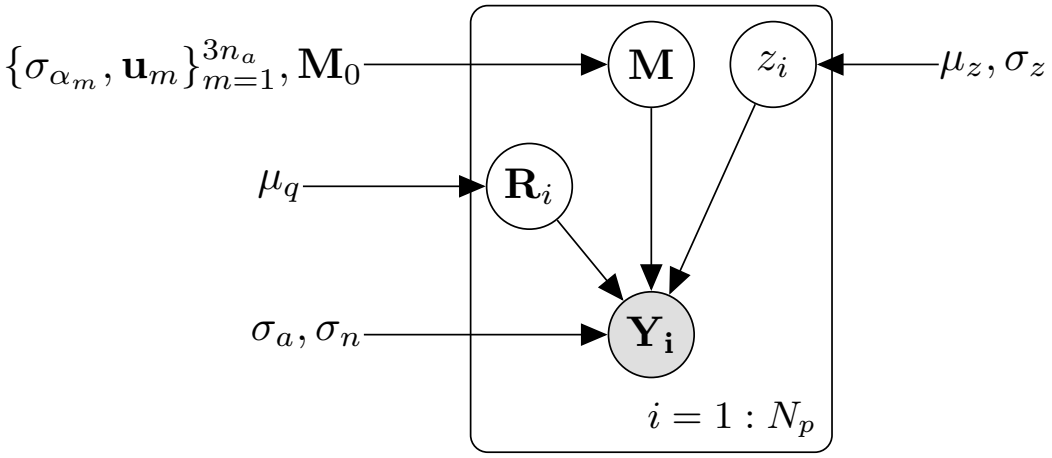

Figure 1: Graphical model of the stochastic physics simulator of cryo-EM image formation. $\mathbf{M} \in \mathbb{R}^{3n_a}$ is the center positions of Gaussian pseudo-atoms, that are additive perturbations to a reference conformation $\mathbf{M}_0$ along a vector fields $\mathbf{u}_m$ defined by the engendecomposition of its Hessian with respect the directional derivatives of each pseudo atom centre. The fixed eigenvector(field) is scaled by a sampled Gaussian with prior variance σ_{α_m}. μ_q is a prior mean and concentration of a unit 4-vector that is converted into a 3D rotation matrix $R_i \in \mathbb{R}^{3\times3}$. z_i is defocus in point spread function of the objective lens of the electron microscope and is sampled from a Gaussian with mean and variance μ_z, σ_z^2. The final measurement $\mathbf{Y_i}$ is a projection onto a 2D gridded array (square pixelated detector) of the Gaussian (of width σ_a) blobs around the atom centres, with additive Gaussian noise with mean zero and variance σ_n^2. N_p particles are iid.

4.2 Stochastic forward model (decoder)

The observed image $\mathbf{Y}$ is simulated by a stochastic forward model, summarized by Figure 1 and the model outlined in Algorithm 1, with further detail given in Appendix-1. To simulate data from the model, one simply returns samples `Y_dist.sample()`, as the observe statement `pyro.sample('noise',Y_dist, obs=...)` is used in variational inference (see section 4.3), and not to simulate data.

4.3 Stochastic variational inference

Stochastic variational inference in the framework provided by the deep probabilistic programming language Pyro (`pyro.infer.SVI`) minimizes the evidence lower bound:

$$\text{ELBO} \equiv \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z})\right] \tag{1}$$

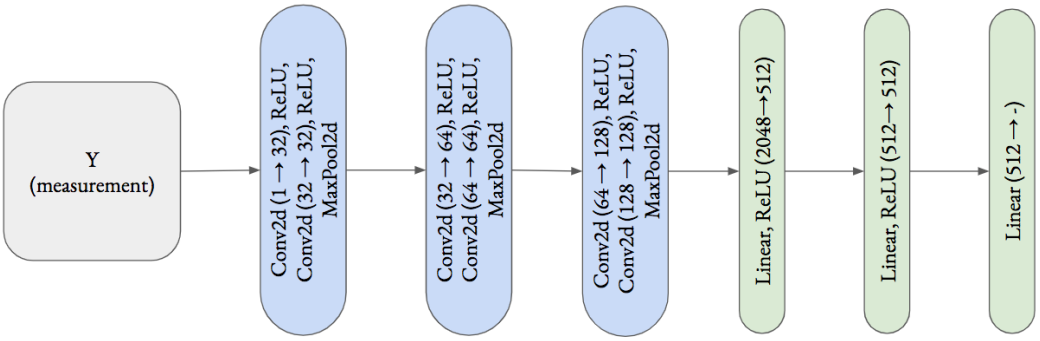

Figure 2: The deep neural net guide / encoder consumes the measured observable and maps this to an interpretable latent space. It contains a neural network for each latent and a chosen distribution. The architecture is three convolution modules (Conv2d, ReLU, Conv2d, ReLU, MaxPool2d), which are flattened and input to three linear layers MLP of size 2048 (for 32 pixel sized input image). The final output size (-) is 2 for the CTF neuran network (μ_z, σ_z), 2 for the normal modes $(\mu_{\alpha_0}, \sigma_{\alpha_0})$), and 10 for the mixture of poses (8 for the direction of $\mu_{q_i},\ i = 1, 2$, 2 for their magnitudes, two for the mixture weights). In total each neural net has 1.6 million parameters.

Algorithm 1 Stochastic forward model of image formation

$\alpha_0 =$ `pyro.sample('enm_scale',` $\mathcal{N}(0, \sigma_{\alpha_0}))$
$\mathbf{M} = \mathbf{M}_0 + \alpha_0 \mathbf{u}_0$
$q =$ `pyro.sample('rotation',` $\mathcal{PN}(\mu_q))$
$\mathbf{R} =$ rotation_from_quaternion(q) ▷ `pytorch3d.transforms.quaternion_to_matrix`
V_{2D} = project$(\mathbf{M}, \sigma_a)$ ▷ fast approx. projection with `torch.sparse_coo_tensor`
$z_i =$ `pyro.sample('defocus',` $\mathcal{N}(\mu_z, \sigma_z))$
CTF = make_ctf(z_i)
`Y_dist` $= \mathcal{N}(V_{2D} \circledast CTF, \sigma_n)$
$\mathbf{Y} =$ `pyro.sample('measurement',Y_dist, obs=...)` ▷ FFT based convolution

Here the stochastic forward model is $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$ is called a `model` in Pyro. The learned posterior is $q_\phi(\mathbf{z})$, is called a `guide`. The guide is the encoder that consumes the measured data and maps it to latent space: $q_\phi(\mathbf{z})$. Here the model does not contain any trainable parameters and $\theta \in \{\}$.

Here the latents are the return of the `pyro.sample` statements. Notice that they are named with a string label in 1-1 between model and guide. The model also contains an additional observe statement: `pyro.sample(...,obs=...)`, which is not included in the guide. This book-keeping a key feature of a probabilistic programming language, and will be done for each trace (execution of the probabilistic program) through the computational graph defined by the model.

Every gradient step in `pyro.infer.SVI` calls the stochastic forward model, and go through the deterministic computations in it. The structure of the guide is shown in Algorithm 2, and further detail, along with the neural network architectures are in outlined in the Appendix-2.

4.4 Training

Training / optimizing the objective with `pyro.infer.SVI` is as simple as passing it a model, guide, optimizer, and loss. Here the `Trace_ELBO` loss is used. Here we used the `pyro.optim.ClippedAdam` optimizer with default parameters, unless otherwise specified.

The parameters in the guide are registered for optimization with `pyro.module`. To "freeze weights" one simply omits this, and they are detached from the computational graph (i.e. `.detach()` in Pytorch). The `pyro.optim.ClippedAdam` optimizer with a learning rate in the range of 10^{-2}–10^{-7}, large batch sizes (50-2000) are used, with 10000 training examples (i.e. no test/train split, only training).

Algorithm 2 Deep inverse guide to learn the posterior

Require: f_z, f_{α_0}, f_q ▷ neural networks
Require: int n_{mix}
$\mu_{z|\text{data}}, \log \sigma_{z|\text{data}} = f_z(\text{data})$
$z_i = \texttt{pyro.sample('defocus',}\mathcal{N}(\mu_{z|\text{data}}, \sigma_{z|\text{data}})\texttt{)}$
$\mu_{\alpha_0|\text{data}}, \log \sigma_{\alpha_0|\text{data}} = f_{\alpha_0}(\text{data})$
$\alpha_0 = \texttt{pyro.sample('enm\_scale',}\ \mathcal{N}(\mu_{\alpha_0|\text{data}}, \sigma_{\alpha_0|\text{data}})\texttt{)}$
$\mu_{q_{1:n_{\text{mix}}|\text{data}}}, \log c_{1:n_{\text{mix}}|\text{data}}, \log w_{1:n_{\text{mix}}|\text{data}} = f_q(\text{data})$ ▷ mean, conc., mixture weights
$\texttt{mix\_dist} = \text{Cat}(w_{1:n_{\text{mix}}|\text{data}})$
$\texttt{comp\_dist} = \mathcal{PN}(\mu_{c_{1:n_{mix}|\text{data}} q_{1:n_{mix}|\text{data}}})$
`qmm_dist = MixtureSameFamily(mix_dist,comp_dist)` ▷ unit quat. mixture model
▷ `torch.distributions.MixtureSameFamily`
$q = \texttt{pyro.sample('rotation',qmm\_dist)}$

5 Data Sets

I generated synthetic data from using the stochastic forward model of same biomolecule as in [18], Aurora A Kinase, but with a smaller box size (32 pixels instead of 64), and only every second alpha carbon (PDB: 1OL5), and thus the protein is coarse grained as 133 pseudo-atoms. Unless otherwise noted, data was generated with the same parameters in the model (its prior): $\mu_z = 20, \sigma_z = 5$ for the CTF defocus, $\mu_{\alpha_m} = 0\ \forall m$, $\sigma_{\alpha_0} = 1$ and $\sigma_{alpha_m} = 0\ \forall m \neq 0$ for the normal modes, and $\sigma_n = 0.06 - 0.3$ for the measurement noise, corresponding to a signal to noise (signal variance / noise variance) of 20. A pose prior $\mu_q = (0, 0, 0, 0)$ was used while the simulated data used μ_q such that $\frac{\mu_q}{||\mu_q||_2} = (1, 0, 0, 0)$ with $||\mu_q||_2$ being various concentrations for the pose ranging 0-1000. For the deterministic projection a Gaussian spread of $\sigma_a = 0.8$ pixels was used and densities were truncated to within a 5^2 pixel patch centred at each atom. Representative simulated data is shown in Figure 3.

6 Experiments

6.1 Stochastic Forward Model

Each gradient step depends on an evaluation of the stochastic forward model, which involves stochastic sampling from distributions characterizing the latents, and deterministic computations defining the distributional parameters. Thus compute efficiency is critical. The compute bottleneck is the projection of the mean atom positions to a 2D array the same size as the simulated image (Table 1). This speed up has been achieved by precomputing the offsets of pixel indeces around each atom, and using a data structure in PyTorch for sparse uncoalesced (repeated indices for when density from pseudo-atoms overlap on the same pixel) tensors on the GPU.

Table 1: Stochastic Forward Model Runtime

Subroutine	**Time (μs/projection)**
normal mode	2-3
Rotation	2-3
3D $\rightarrow$ 2D projection	34-36
CTF	10-13
Shot noise	30-33

Runtime for $n_a = 133$ pseudo-atoms, batch size of 1000 (projections/batch), $n_{\text{pix}} = 128$, $\sigma_a = 2$ (pix), 3D$\rightarrow$2D projection truncation box length `n_trunc`$= 6\sigma_a = 12$ (pix).

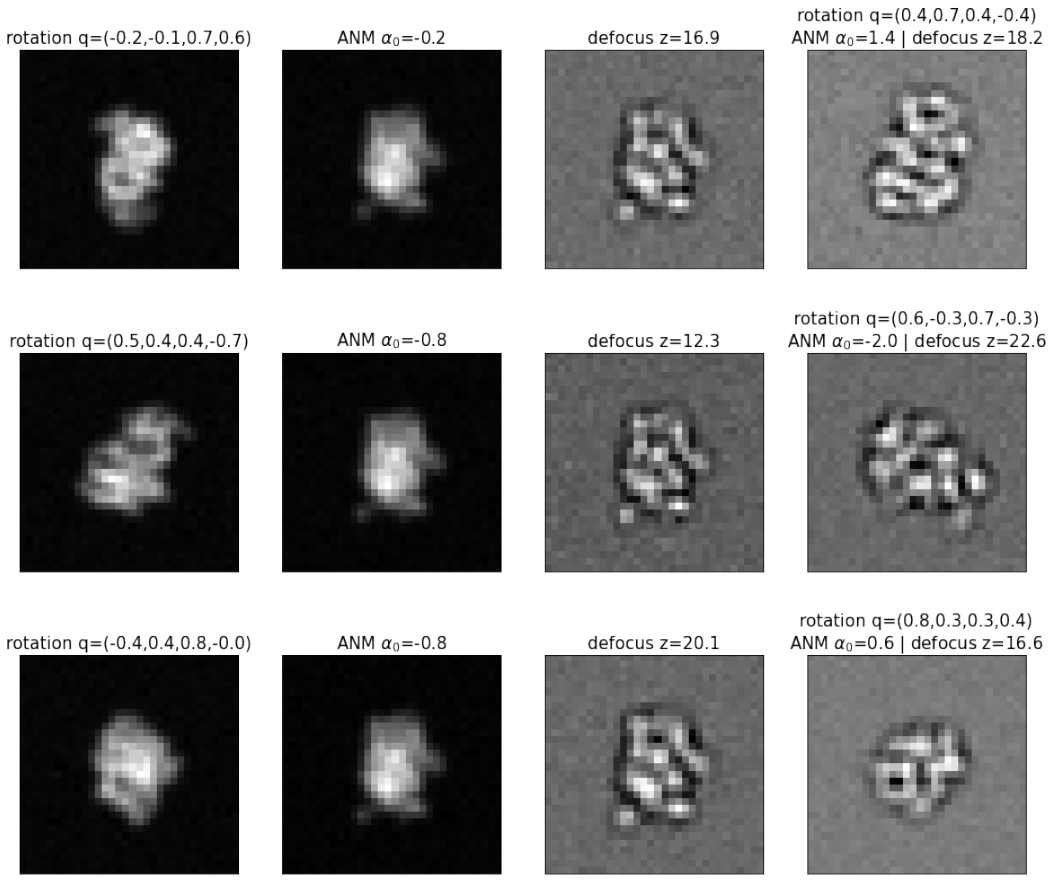

Figure 3: Simulated data with ground truth labels. Three examples are shown for simulated data with only one latent (columns 1-3), or all three applied (last column).

6.2 Inferring individual latents

As a proof of principle, synthetic data was generated with only one latent variable (either CTF, normal mode, or pose) and then the posterior was characterized through training. Learning the CTF (with no normal mode perturbation and fixed pose) and the normal mode perturbation (with CTF $= 1$, fixed pose) was feasible to a high correlation of the sampled latent to the ground truth value (pearson correlation > 0.9).

In contrast, characterizing the pose (with CTF $= 1$, no normal mode) ran into numerical stability issues scoring `pyro.distributions.ProjectedNormal` (Figure 4) and the Appendix (Figure A1). While training, scoring the Projected Normal resulted in `nan` being returned from the `ProjectedNormal.log_prob`, which then propagated to the neural network weights, and then the returned distributional parameters for the ProjectedNormal that were all `nan` (every element). The following aggrevated this issue: less shot noise, a more concentrated ProjectedNormal prior, larger step size, smaller batch size. In order to overcome this, the returned (log) concentrations from f_q were clamped in the guide before being making the mixture of Projected Normals. The concentration was clamped such that any sample should return a finite `log_prob`, which turned out to be ~ 4.13, which is still fairly spread out (Appendix Figure A1). The posterior distribution was therefore prevented from becoming more sharply peaked, and samples are spread out such that the reconstruction would be inaccurate. I am currently in contact with he Pyro development team and working on a numerically stable fix (Appendix 4).

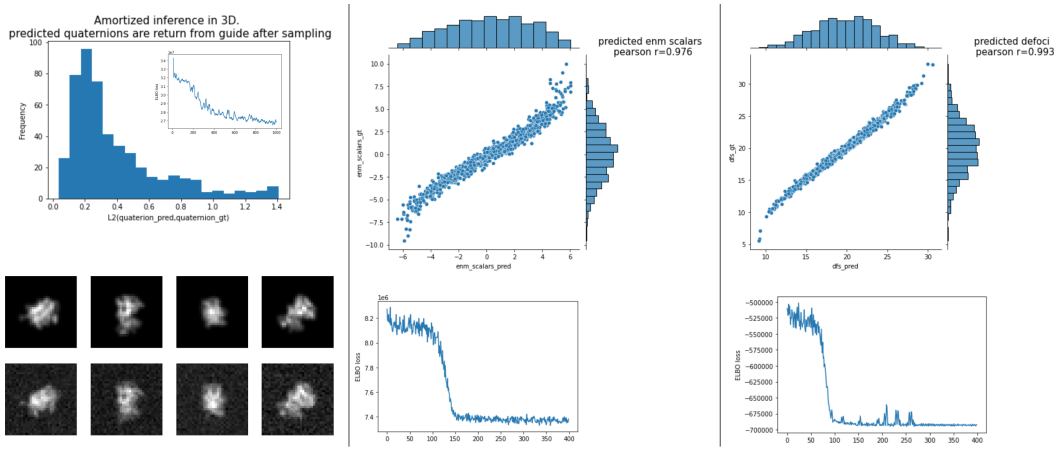

Figure 4: Samples from the individually learned latents are close to their ground truth values: left column pose, middle normal mode, right CTF defocus. For pose prediction, the noise free projection from the predicted pose (mean rotation for the maximal mixture component) is shown above the simulated image (with noise).

6.3 Inference of multiple latents

At fixed pose (i.e. no rotation applied in the model or guide, and the parameters of f_q not optimized) the CTF defocus and normal mode component could be inferred accurately (Figure 5). A learning schedule was used with a learning rate of 0.0005 where f_{α_0} was optimized for 20 epochs (batch size 500, 400 gradient steps), until the predicted normal modes had high correlation (Figure 5, left panel). It was important to not optimize f_z at this point to avoid the CTF from becoming poorly characterized - the known μ_z in the guide keeps the fluctuating around reasonable values (Figure 5, left panel). Then the weights of f_{α_0} were frozen and f_z was optimized for a similar 20 epochs, after which is also had high correlation (Figure 5, right panel).

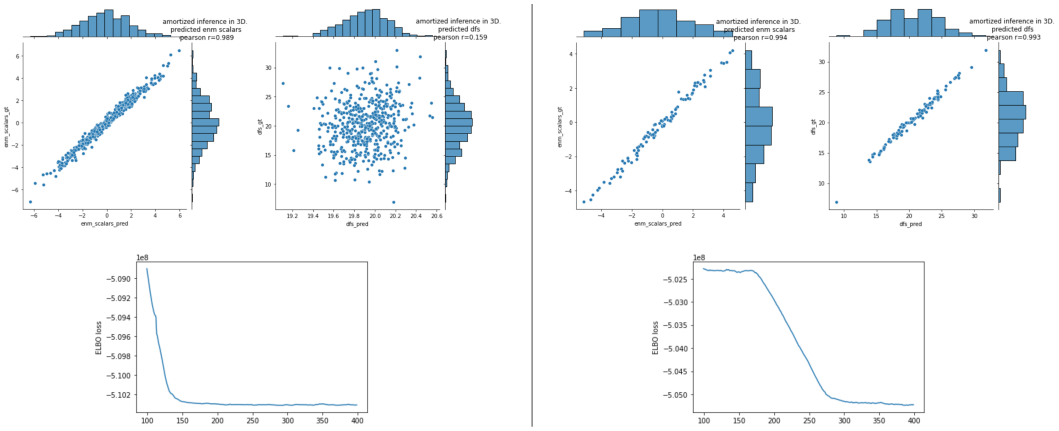

Figure 5: All latents can be jointly inferred from the data. Samples from the learned posterior are close to their ground truth values, and spread with increased noise. Note that in the right panel the have less points plotted that the left due to low GPU memory issues after training.

7 Conclusion and Future Work

The overarching goal of this physics aware approach is to incorporate information from past characterization of the latents (e.g. whole micrograph CTF parameter estimates, noise level, atomic model estimate or past published structure) in a principled manner, and jointly infer a latent space that is

disentangled through an interpretable forward model that is a physics simulation of the imaging process.

Here I have shown for the first time inference of the CTF defocus using a deep neural network, used a mixture model of 4 dimensional Projected Normals to characterize the posterior of 3D pose, and jointly predicted normal mode based conformational heterogenity and CTF. This method can be readily extended to incorporate more complex forward models, include conditioning in the guide, and alternative neural network architectures. However some issues first need to be overcome.

In order to infer the pose together with the CTF defocus and component along the normal mode(s), the numerical instabilities in the `log_prob` of the Projected Normal should be overcome in order to more tightly characterized the posterior of pose. Furthermore, another distribution altogether could be used. For instance previous studies have used s2_s2 encoding of rotations, one future area to explore is sampling $\mathbb{R}^6$ with Gaussian distributions, and converting this to a rotation as in [18, 13, 27].

Using (multiple) ANM normal modes reduces the degrees of freedom of the atomic centers to scalars along fixed vector fields, and the relative ranking of the modes allows only a few degrees of freedom to express some physically plausible conformational heterogenity. Here only one has been used as a proof in principle, but could be extended to several modes, which typically capture some dominating breathing motions [5] of proteins in thermal equilibrium, but not things like large rigid body rototranslations or subtle rearangement of local catalytic sites [14, 25, 3]. This is arguably more physically interpretable than the way the conformational heterogeneity was output in [18] by the VAE decoder that output a final linear layer $\in \mathbb{R}^{9 \cdot N_{\text{res}}}$. The nine (9) components for each residue are used to define the translation ($\mathbb{R}^3$) and rotation ($\mathbb{R}^6 \rightarrow \mathbb{R}^{3\times3}$ through Gram-Schmidt orthogonalization).

Some disadvantages of the normal modes are (1) if the reference atom position $\mathbf{M}_0$ is updated, the normal modes need to be recomputed with an expensive engendecomposition (e.g. SVD on hundreds to thousands of nodes). (2) If pseudo-atoms are not in proximity to a cluster of other psdueo-atoms the eigenvector at that location has a very large magnitude - e.g. flexible loops. In practice these can be and often are "trimmed" in a pre-processing step, or anchored as a rigid body [21]. This suggests that the low modes will tend to express flexible and floppy regions of biomolecules which have small projected potential associated with them and therefore may be hard to learn because the sigmal for them in the projection is small. It may be possible to learn the normal modes themselves, by predicting an additive pertubation vector field as an output from a neural network.

In conclusion, by using a probabilistic programming framework such as Pyro, I am striving to achieve a rapid development cycle that easily allows researchers to extend the forward model to higher levels of theory and more physically accurate measurement/noise models; to model conformational heterogeneity with an appropriate level of granularity/coarseness; to input domain knowledge of electron optics priors and priors on atomic conformations. These sources of inductive bias should make the optimization more favourable, and inference more interpretable. For structural biologists, interpretability means representations of the microscopic formal causes in a familiar format that is shared between imaging and spectroscopic modalities, such as atomic conformations as the 50+ years of the Protein Data Bank attests to [2, 1, 8, 24]. The payoff of interpretability is to more deeply engage with the angstrom level mysteries of living nature that challenge our scientific imaginations to respond with principled rigour [11].

References

[1] A celebration of structural biology. *Nature Methods*, 18(5):427–427, may 2021.

[2] Happy anniversary, PDB! *Nature Structural Molecular Biology*, 28(5):399–399, may 2021.

[3] Ivet Bahar, Timothy R. Lezon, Lee-Wei Yang, and Eran Eyal. Global Dynamics of Proteins: Bridging Between Structure and Function. *Annual Review of Biophysics*, 39(1):23–42, apr 2010.

[4] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

[5] Ken Dill, Robert L. Jernigan, and Ivet Bahar. *Protein Actions*. Garland Science, New York, NY : Garland Science, Taylor Francis Group, LLC, [2017] |, sep 2017.

[6] Ken A. Dill and Sarina Bromberg. *Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience*. Garland Science, 2010.

[7] Robert M Glaeser, Eva Nogales, and Wah Chiu, editors. *Single-particle Cryo-EM of Biological Macromolecules*. IOP Publishing, may 2021.

[8] David S. Goodsell. Art as a tool for science. *Nature Structural Molecular Biology*, 28(5):402–403, may 2021.

[9] Daniel Hernandez-Stumpfhauser, F. Jay Breidt, and Mark J. van der Woerd. The general projected normal distribution of arbitrary dimension: Modeling and Bayesian inference. *Bayesian Analysis*, 12(1):113–133, 2017.

[10] Grant J. Jensen, editor. *Methods in Enzymology, volume 482: Cryo-EM, Part B: 3-D Reconstruction*. Academic Press, 2010.

[11] Dorothee Kern. From structure to mechanism: skiing the energy landscape. *Nature Methods*, 18(5):435–436, may 2021.

[12] Alexander Lavin, Hector Zenil, Brooks Paige, David Krakauer, Justin Gottschlich, Tim Mattson, Anima Anandkumar, Sanjay Choudry, Kamil Rocki, Atılım Güneş Baydin, Carina Prunkl, Brooks Paige, Olexandr Isayev, Erik Peterson, Peter L. McMahon, Jakob Macke, Kyle Cranmer, Jiaxin Zhang, Haruko Wainwright, Adi Hanuka, Manuela Veloso, Samuel Assefa, Stephan Zheng, and Avi Pfeffer. Simulation Intelligence: Towards a New Generation of Scientific Methods. 2021.

[13] Axel Levy, Frédéric Poitevin, Julien Martel, Youssef Nashed, Ariana Peck, Nina Miolane, Daniel Ratner, Mike Dunne, and Gordon Wetzstein. CryoAI: Amortized Inference of Poses for Ab Initio Reconstruction of 3D Molecular Volumes from Real Cryo-EM Images. mar 2022.

[14] Jianpeng Ma. Usefulness and limitations of normal mode analysis in modeling dynamics of biomolecular complexes. *Structure*, 13(3):373–380, 2005.

[15] Suvrajit Maji and Joachim Frank. What is in the black box? – A perspective on software in cryoelectron microscopy. *Biophysical Journal*, 120(20):4307–4311, 2021.

[16] Youssef S. G. Nashed, Frederic Poitevin, Harshit Gupta, Geoffrey Woollard, Michael Kagan, Chun Hong Yoon, and Daniel Ratner. CryoPoseNet: End-to-End Simultaneous Learning of Single-particle Orientation and 3D Map Reconstruction from Cryo-electron Microscopy Data. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, volume 1, pages 4049–4059. IEEE, oct 2021.

[17] Ali Punjani, John L. Rubinstein, David J. Fleet, and Marcus A. Brubaker. CryoSPARC: Algorithms for rapid unsupervised cryo-EM structure determination. *Nature Methods*, 14(3):290–296, feb 2017.

[18] Dan Rosenbaum, Marta Garnelo, Michal Zielinski, Charlie Beattie, Ellen Clancy, Andrea Huber, Pushmeet Kohli, Andrew W. Senior, John Jumper, Carl Doersch, S. M. Ali Eslami, Olaf Ronneberger, and Jonas Adler. Inferring a Continuous Distribution of Atom Coordinates from Cryo-EM Images using VAEs. *arXiv*, pages 1–15, 2021.

[19] Sjors H.W. Scheres. RELION: Implementation of a Bayesian approach to cryo-EM structure determination. *Journal of Structural Biology*, 180(3):519–530, 2012.

[20] Amit Singer. Mathematics for cryo-electron microscopy. 2018.

[21] Florence Tama, Florent Xavier Gadea, Osni Marques, and Yves-Henri Sanejouand. Building-block approach for determining low-frequency normal modes of macromolecules. *Proteins: Structure, Function, and Genetics*, 41(1):1–7, oct 2000.

[22] Guang Tang, Liwei Peng, Philip R Baldwin, Deepinder S Mann, Wen Jiang, Ian Rees, and Steven J Ludtke. EMAN2: an extensible image processing suite for electron microscopy. *Journal of structural biology*, 157(1):38–46, jan 2007.

[23] Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood. An Introduction to Probabilistic Programming. *arXiv*, pages 1–221, 2018.

[24] Jasmine Y. Young, John Berrisford, and Minyu Chen. wwPDB biocuration: on the front line of structural biology. *Nature Methods*, 18(5):431–432, 2021.

[25] She Zhang, James M Krieger, Yan Zhang, Cihan Kaya, Burak Kaynak, Karolina Mikulska-Ruminska, Pemra Doruker, Hongchun Li, and Ivet Bahar. ProDy 2.0: increased scale and scope after 10 years of protein dynamics modelling with Python. *Bioinformatics*, 37(20):3657–3659, oct 2021.

[26] Ellen D. Zhong, Adam Lerer, Joseph H. Davis, and Bonnie Berger. Exploring generative atomic models in cryo-EM reconstruction. *arXiv*, pages 1–13, jul 2021.

[27] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:5738–5746, 2019.

Appendix: Physics aware joint inference for the cryo-EM inverse problem: normal modes, global 3D pose and CTF defocus.

Geoffrey Woollard
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
gw@cs.ubc.cs

1 Stochastic forward model

1.1 Conformational heterogeneity

The conformational heterogeneity can be understood from the perspective of an energy model physically inspired by each pseudo-atom being a "ball" attached by "springs" to other pseudo-atoms [2].

$$\mathbb{P}(\mathbf{M}) = Z^{-1} \exp[-\beta U(\mathbf{M})] \tag{1}$$

$$U_{\text{anm}} = \frac{\gamma}{2} \sum_{ij} (r_{ij} - r_{0,ij})^2 \tag{2}$$

The anisotropic network model has energy U_{anm}, where r_{ij} is the distance between pseudo-atom pair ij in the sample $\mathbf{M}$, $r_{0,ij}$ is the corresponding reference distance in $\mathbf{M}_0$, and γ is a spring constant. The second derivative elements of the $3n_a \times 3n_a$ Hessian has a convenient analytical form with 3×3 symmetric ij submatrices given by

$$\mathbf{H}_{ij} = \frac{\gamma}{r_{ij}^2} \begin{bmatrix} x_{ij}^2 & x_{ij}y_{ij} & x_{ij}z_{ij} \\ x_{ij}y_{ij} & y_{ij}^2 & y_{ij}z_{ij} \\ x_{ij}z_{ij} & y_{ij}z_{ij} & z_{ij}^2 \end{bmatrix} \tag{3}$$

And the ii diagonal submatrices given by the row/column sum $\mathbf{H}_{ii} = \sum_{j \neq i} \mathbf{H}_{ij}$. The anisotropic network model is anisotripic in the sense that each xyz direction has its own Hessian component and can be different–this makes it directional. The probability is then approximated by a second order Taylor expansion about a reference pseudo-atomic configuration $\mathbf{M}_0$:

$$U(\mathbf{M}) = U(\mathbf{M}_0) - \frac{1}{2}(\mathbf{M} - \mathbf{M}_0)^T \mathbf{H} (\mathbf{M} - \mathbf{M}_0) \tag{4}$$

The eigendecomposition of $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{\mathbf{T}}$, enables to project any pseudo-atom configuration $\mathbf{M}$ onto components of $\mathbf{U}$, because $\alpha_m = \mathbf{u}_m^T(\mathbf{M} - \mathbf{M}_0)\, \forall m$. Thus $\mathbf{M}$ is a deterministic change of basis to the set $\{\alpha_m\}_1^{3n_a}$.

Preprint. Under review.

This exponential pdf reduces the probability to a diagonal multivariate Gaussian through the orthogonality of the basis.

$$\mathbb{P}(\mathbf{M}) = \mathbb{P}(\{\alpha_m\}) \tag{5}$$

$$= (\det[2\pi\mathbf{\Lambda}])^{-1/2} \prod_m \exp -\beta \frac{\alpha_m^2}{\lambda_m} \tag{6}$$

$$= \prod_m \mathbb{P}(\alpha_m) \tag{7}$$

Thus a sample of pseudo-atomic positions is obtained by additively perturbing the mean pseudo-atomic positions $\mathbf{M_0} \in \mathbb{R}^{3n_a}$ by a perturbation vector, $\mathbf{u}_{\text{perturb}} = \sum_m \alpha_m \mathbf{u_m} \in \mathbb{R}^{3n_a}$, where each α_m is sampled from a Gaussian distribution, and each elastic network mode are fixed for constant $\mathbf{M_0}$.

$$\mathbf{M} = \mathbf{M_0} + \mathbf{u}_{\text{perturb}} \tag{8}$$

In the simplified model employed here, $\mathbf{u}_{\text{perturb}}$ is restricted the single lowest mode $\mathbf{u}_0$, ranked by eigenvalue $\lambda_0 < \lambda_m,\ \forall\ m \neq 0$, and thus $\mathbf{M}$ is being sampling according to the distribution $p(\mathbf{M}|\sigma_{\alpha_0}, \{\sigma_{\alpha_m} = 0\}_{m\neq 0}) = p(\alpha_0|\alpha_1 = 0, ..., \alpha_{n_a-1} = 0) \propto \mathcal{N}(\alpha_m|0, \sigma_{\alpha_m} = \frac{\lambda_0}{2\beta})$

Thus I explicitly compose the Hessian $\mathbf{H}$ of a reference conformation $\mathbf{M_0}$, compute its low mode eigenvectors and values, and keep this precomputed in memory. During stochastic simulation I sample a Gaussian scalar and additively perturb the reference conformation. Thus stochastic sampling of $\mathbf{M}$ is as fast as sampling Gaussians, scaling their corresponding eigenvectors, summing them to one eigenvector, and adding this perturbation to the reference conformation $\mathbf{M}_0$.

1.2 Global rotation

A global rotation is sampled from the uniform distribution on SO(3), and the rotated pseudo-atoms are projected along the imaging axis: $\mathbf{R_q};\ q \sim \mathcal{PN}[\mu_q]$, where q is a unit quaternion and which is sampled by the Projected Normal distribution [4]. In contrast to voxel based rotations, and Fourier slices, the potential is parametrized by pseudo-atoms in $\mathbb{R}^3$, where each pseudo-atomic xyz poisition $\mu_k \in \mathbb{R}^3$ are simply rotated through fast vectorized matrix multiplication: $\mathbf{M} \rightarrow \mathbf{RM}$ with $\{\mathbf{RM}\}_k = \mathbf{R}\mu_k$

$$\mathbf{RM} = \begin{bmatrix} \mathbf{R}\mu_1 \\ \mathbf{R}\mu_2 \\ ... \\ \mathbf{R}\mu_{n_a} \end{bmatrix} \tag{9}$$

1.3 Projection

The 2D projected potential is the integral of the 3D potential, along the direction of the imaging axis z: $V_{2D} = \int dz\ V(x, y, z)$. The potential is assumed to be an additive sum of non interacting pseudo-atomic densities, via a Gaussian kernel: $V(x, y, z) = \sum_{k=1}^{n_a} \exp[\frac{-||(x,y,z)^T - \mathbf{R}\mu_k]||_2}{2\sigma_a^2}]$. Since the size of pixels on the order σ_a, and a Gaussian is extremely close to zero after a few σ_a. In particular $\exp[\frac{-(n\sigma_a)^2}{2\sigma_a^2}] = \exp(-n^2/2) = 0.01$ for $n = 3$ or 0.0003 for $n = 4$, and so I truncate the Gaussian to zero after several σ. I leverage this for a GPU sped up projection deterministic linking function, which can project tens of thousands of pseudo-atoms to a 512^2 sized image in hundreds of μ seconds, and which does not experience a memory bottleneck for large box sizes.

1.4 Microscope optics

The microscope's point spread function, i.e. the contrast transfer function, CTF $= \sin[2\pi\chi]$, is here assumed to be a circular symmetric polynomial in frequency $\mathbf{k}$, although our method can be extended to higher order abberations [3].

$$\chi = \frac{-z\lambda k^2}{2} - \frac{C_s\lambda^3 k^4}{4}. \tag{10}$$

The wave length λ and spherical abberation C_s are assumed to be known, global, and fixed (distributed by the delta distribution). The defocus z is distributed by a Gaussian centred at a known mean z_0, given, for example from whole micrograph CTF estimation: $z \sim \mathcal{N}(\mu_z, \sigma_z)$.

The CTF is applied in Fourier space via the Fast Fourier Transform (FFT). This means that each simulation of the forward model involves an FFT on the simulated projection, constructing the CTF given the sample of the defocus, element wise multiplication, and an inverse Fast Fourier transform.

1.5 Measurement

The distributional parameters to the final observed image $\mathbf{Y}$ is the 2D projected potential, $V_{2D} = \int dz\, V(x, y, z)$, convoluted with a circular symmetric sinusoidal CTF $= \sin[2\pi\chi]$.

The effects of the detector (shot noise, point spread function), electronic read out noise, solvent noise, sample damage and unmodeled density ("stuff") are all modelled by additive white Gaussian noise.

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{Y}|\mu_{\mathbf{Y}} = V_{2D} \circledast \text{CTF}, \sigma_n) \tag{11}$$

1.6 Probabilistic structure

Each measurement Y_i is i.i.d with global rotation $\mathbf{R}_i$, defocus scalar z_i, and conformational heterogeneity scalar α_i being i.i.d. to each other, and factorizing as

$$\begin{aligned}
\mathbb{P}(\text{data}|\text{model}) &= \mathbb{P}(\{\mathbf{Y}_i\}|\{\alpha_i\}, \{\mathbf{R}_i\}, \{z_i\}) && (12)\\
&= \prod_i \mathbb{P}(\mathbf{Y}_i|\alpha_i, \mathbf{R}_i, z_i) && (13)\\
&= \prod_i \mathbb{P}(\mathbf{Y}_i|\alpha_i, \mathbf{R}_i, z_i) && (14)\\
&= \prod_i \mathbb{P}(\mathbf{Y}_i|\mu_{\mathbf{Y}}, \sigma_n)\mathbb{P}(\alpha_i)\mathbb{P}(\mathbf{R}_i)\mathbb{P}(z_i) && (15)
\end{aligned}$$

Such a forward model yields $N_p = 10^4 - 10^6$ i.i.d simulated particles.

2 Inverse guide (encoder)

In Pyro's stochastic variational inference setting, the model (decoder) can express a stochastic forward model of image formation with arbitrary control flow, and this physics-awareness makes it interpretable. The guide can contain rich distributions to characterize the posterior distribution of the unobserved latent variables in the computational graph, and learn the parameters of the distributions with deep architectures. Including learnable parameters in the model generalized amortized inference to model learning. Choosing the architecture of the guide, and the control flow of the inverse function corresponds to a choice on the inverse computational graph. For instance, one can first sample the microscope parameters, and then incorporate this sample and the observed data into another deep net that samples the pose, and then incorporate the sample of the microscope parameters and pose into another deep net that samples conformational heterogeneity. This was done for pose and conformation in [7] When possible, the observed data can be modified to "undo" the effect of latent variable, e.g. unshifting or un-rotating an in-plane shift of 2D rotation, deconvolution of a point spread function.

In Pyro's `pyro.infer.SVI` the `guide` passes the samples to the `model`, which just uses them and evaluates the joint probability–the likelihood and prior under the distribution in the `model`. So the `pyro.sample` statements in the model should be thought about as instructions to "go to guide and get sample".

In the guide a `pyro.sample` statement is needed for every `pyro.sample` statement in the model, with the names, but not necessarily the type of distribution, lining up 1-1. There are no observe statements in the guide. The distribution type in the guide is chosen so that the samples from it fall in the support of the model's distribution. So, for example, a Gaussian distribution in the guide would not be a good match for a uniform distribution in the model, since a sample from the guide outside

of the support of the Uniform would be impossible and the `.log_prob` for a zero probability event would hypothetically be negative infinity.

I use a mixture of `ProjectedNormal` for the rotation. A multimodel distribution of the pose posterior would be possible to characterize with a mixture distribution, and if there were many modes (6-12 or higher is not unseen in membrane proteins and bacterial injection systems) for higher symmetry specimens (discussed in [6, 5]) the mixture model could have increased components to better characterize higher symmetries.

3 Training

Other versions of the loss are available in Pyro besides `Trace_ELBO`, for when the distributions aren't reparametrizable (do not have an `.rsample()` method) and require approaches like REINFORCE to esimate the gradient. See discussion `http://pyro.ai/examples/svi_part_iii.html`).

4 Projected normal numerical stability

During training the returned distributional parameters from neural networks characterizing the projected normal distribution frequently returned a tensor of all `nan` valuess. Upon further investigation and discussion with the Pyro development team [1], the cause is likely to be a numerically unstable implementation of the `log_prob`. Uniform samples from $SO(3)$ have a `log_prob` of `nan` values when the distribution they are scored under has concentration > 4.13, which is fairly diffuse over $SO(3)$ as shown in Figure S1.

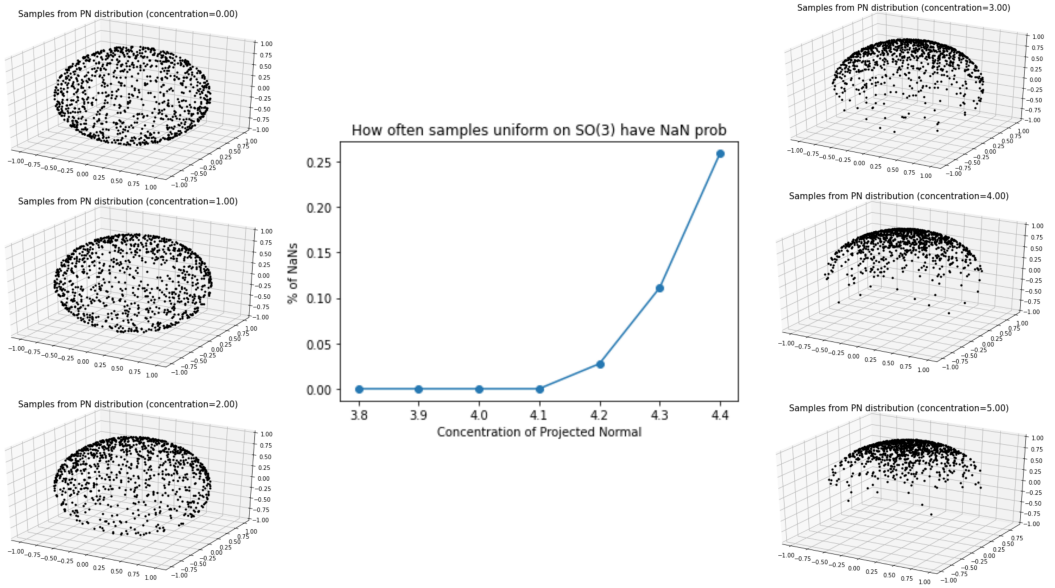

Figure 1: Rotations were projected onto the sphere along the vertical axis: $\mathbf{R}(0, 0, 1)^T$ at various concentrations (left and right panels; 1000 points plotted). The `log_prob` of Projected Normal distributions of various concentrations $(3.8, 3.9, 4, 4.1, 4.2, 4.3, 4.4)$ were computed for 10^6 samples (from a Projected Normal with zero concentration) and the percentage of `nan`s returned was (0, 0, 0, 0, 0.0275, 0.1111, 0.2588)% respectively (center panel).

The ELBO loss term itself involves scoring samples (from the guide) with the prior distribution in the model (zero concentration) and the posterior distribution in the guide (learned concentration). While this term should not evaluate uniform samples by a concentrated distribution, the gradient term is a likely candidate and is consistent with this issue arising during training even when the learned posterior concentration is clamped to be < 4.13, for example at 4. See [1] ref for a similar issue, and

[1] `https://forum.pyro.ai/t/svi-nans-from-guide-when-trace-elbo-drops/4102/2`

comments on the numerical instabilities in the modified Bessel function. The numerical instabilities are likely to arise from the `para_part` in `_log_prob_4` [2].

Listing 1: Pyro Projected Normal: Suspected numerical instability in `pyro.ProjectedNormal._log_prob_4`

```
para_part = (
                (2 + t2) *
                t2.mul(-0.5).exp() / (2 * math.pi) ** 0.5 +
                t * (3 + t2) *
                (1 + (t * 0.5**0.5).erf()) / 2
            ).log()
```

This is

$$\log[f_{sq} * f_{\exp} + f_{cub} * f_{\text{erf}}] \tag{16}$$

where

$$f_{sq} = 2 + t^2 \tag{17}$$

$$f_{\exp} = \frac{1}{\sqrt{2\pi}} \exp[\frac{-t^2}{2}] \tag{18}$$

$$f_{cub} = t(3 + t^3) \tag{19}$$

$$f_{\text{erf}} = \frac{1}{2}(1 + \text{erf}[t/\sqrt{2}]) \tag{20}$$

Scoring low probability events corresponds to "large" negative t, where large is $t < -4.10$. Taking a closer look into the numerical pieces shows low probability events occur when the additive terms cancel each other out, $f_{cub} * f_{\text{erf}} = -f_{sq} * f_{\exp}$, and we try to $\log$ a negative number. However, the integral that gives rise to this, $\frac{1}{\sqrt{2\pi}} \int_0^\infty dx\ x^3 \exp[-(x-t)^2/2]$ should be non-negative and only approach zero as $t \to -\infty$, when we handle the asymptotic analytically. The $f_{cub} * f_{\text{erf}}$ piece lacks numerical stability for large negative t likely because f_{cub} sharply increases while f_{erf} decreases, and their multiplication is too negative, making the log negative.

After bringing this to the attention of the Pyro development team, and providing evidence in some worked examples illustrating numerical instability [3] the pull request was submitted by Fritz Obermeyer to "Numerically stabilize ProjectedNormal.log_prob() via logaddexp" [4]. However, this initial fix did not work for negative t, and further work needs to be done. Numerically stabilizing it could using a `logsumexp` implementation that can factor out a common multiplicative factor fo f_{cub} and f_{erf} to make their produce more stable, and an additive factor of $f_{sq} f_{\exp}$ and $f_{cub} * f_{\text{erf}}$ to make taking the log more numerically stable.

References

[1] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 2:856–865, 2018.

[2] Ken Dill, Robert L. Jernigan, and Ivet Bahar. *Protein Actions*. Garland Science, New York, NY : Garland Science, Taylor Francis Group, LLC, [2017] |, sep 2017.

[3] Robert M Glaeser, Eva Nogales, and Wah Chiu, editors. *Single-particle Cryo-EM of Biological Macromolecules*. IOP Publishing, may 2021.

[4] Daniel Hernandez-Stumpfhauser, F. Jay Breidt, and Mark J. van der Woerd. The general projected normal distribution of arbitrary dimension: Modeling and Bayesian inference. *Bayesian Analysis*, 12(1):113–133, 2017.

[2] `https://docs.pyro.ai/en/dev/_modules/pyro/distributions/projected_normal.html`

[3] `https://colab.research.google.com/gist/geoffwoollard/7422a99189bb26a1189f053cc39b1bf0/pyro_projectednormal_numerical_stability.ipynb`

[4] `https://github.com/pyro-ppl/pyro/pull/3071`

[5] Axel Levy, Frédéric Poitevin, Julien Martel, Youssef Nashed, Ariana Peck, Nina Miolane, Daniel Ratner, Mike Dunne, and Gordon Wetzstein. CryoAI: Amortized Inference of Poses for Ab Initio Reconstruction of 3D Molecular Volumes from Real Cryo-EM Images. mar 2022.

[6] Ruyi Lian, Bingyao Huang, Liguo Wang, Qun Liu, Yuewei Lin, and Haibin Ling. End-to-end orientation estimation from 2D cryo-EM images. *Acta Crystallographica Section D Structural Biology*, 78(2):174–186, 2022.

[7] Dan Rosenbaum, Marta Garnelo, Michal Zielinski, Charlie Beattie, Ellen Clancy, Andrea Huber, Pushmeet Kohli, Andrew W. Senior, John Jumper, Carl Doersch, S. M. Ali Eslami, Olaf Ronneberger, and Jonas Adler. Inferring a Continuous Distribution of Atom Coordinates from Cryo-EM Images using VAEs. *arXiv*, pages 1–15, 2021.