EQUIVARIANT AMORTIZED INFERENCE OF POSES FOR CRYO-EM

Larissa de Ruijter *
QUVA Lab, University of Amsterdam
larissaderuijter@kpnmail.nl

Gabriele Cesa
Qualcomm AI Research, Amsterdam†
QUVA Lab, AMLab, University of Amsterdam
gcesa@qti.qualcomm.com

ABSTRACT

Cryo-EM is a vital technique for determining 3D structure of biological molecules such as proteins and viruses. The cryo-EM reconstruction problem is challenging due to the high noise levels, the missing poses of particles, and the computational demands of processing large datasets. A promising solution to these challenges lies in the use of amortized inference methods, which have shown particular efficacy in pose estimation for large datasets. However, these methods also encounter convergence issues, often necessitating sophisticated initialization strategies or engineered solutions for effective convergence. Building upon the existing cryoAI pipeline, which employs a symmetric loss function to address convergence problems, this work explores the emergence and persistence of these issues within the pipeline. Additionally, we explore the impact of equivariant amortized inference on enhancing convergence. Our investigations reveal that, when applied to simulated data, a pipeline incorporating an equivariant encoder not only converges faster and more frequently than the standard approach but also demonstrates superior performance in terms of pose estimation accuracy and the resolution of the reconstructed volume. Notably, D_4-equivariant encoders make the symmetric loss superfluous and, therefore, allow for a more efficient reconstruction pipeline.

1 INTRODUCTION

Cryo-electron microscopy (cryo-EM) has emerged as an crucial technique in molecular biology and chemistry, enabling the determination of macro-molecular structures such as proteins. In cryo-EM, particle samples are frozen in a thin layer of vitreous ice and exposed to an electron beam. The interaction between electrons and the sample's electrostatic potential scatters electrons in patterns that reflect the molecular structure. This results in multiple noisy two-dimensional projections of the particles in random orientations, which are then used to reconstruct the molecular structure.

The reconstruction process presents significant challenges, including the estimation of unknown poses and the high noise levels Singer & Shkolnisky (2011); Singer & Sigworth (2020). Traditional methods employ iterative approaches that, in each step, require extensive searches over the space of poses *for each datapoint* and a refinement of the molecular structure estimate (e.g. maximum-likelihood expectation-maximization (ML-EM) implemented by SOTA softwares like RELION Scheres (2012) and cryoSPARC Punjani et al. (2017)); however, this per-image search doesn't scale well with the increasing datasets sizes due its computational demand Levy et al. (2022a).

Recently, deep learning approaches have shown promise in addressing these challenges, giving rise to a new category of reconstruction methods based on *amortized inference* Giri et al. (2023). These methods learn a parametrized function, typically within a framework employing (variational) autoencoder architectures, to predict particle poses from images, shifting the computational burden to the learning phase. Although this reduces the need for per-image pose estimation, these solutions often face convergence issues, requiring careful engineering to achieve satisfactory performance.

*Work done during internship at QUVA Lab.
†Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Contributions We introduce *equivariant* amortized inference of poses for cryo-EM. This method uses the property that images that differ by some in-plane rotation or mirroring have associated poses that undergo a similar transformation. In the amortized inference framework, we can exploit this prior knowledge of the geometry of the problem by making the parameterized function *equivariant* with respect to rotations and/or reflections. Using an equivariant function has the following benefits:

- *faster generalization = faster reconstruction*: once it learns to predict the pose of an image correctly, it immediately generalizes to rotations and reflection of that particular image. This typically also leads to faster convergence and, thus, to faster reconstruction here.
- *geometrical consistency = reduced convergence issues*: it is forced from the start to represent subsets of datapoints according to the correct underlying geometric structure, potentially reducing or preventing the convergence issues discussed earlier.

2 IMAGE FORMATION MODEL

We can model the electrostatic potential of a molecule as a 3D density, $V\colon \mathbb{R}^3 \to \mathbb{R}$. Each molecule, frozen in ice, assumes a random orientations represented by a rotation $R_i \in \mathrm{SO}(3)$. The image is formed by electrons that are scattered by interacting with this potential, hitting a detector. This is modeled as a projection:

$$P_{R_i}[V]\colon (x, y) \to \int_z V(R_i^\top (x, y, z)^\top)\, dz. \tag{1}$$

Before the electrons hit the detector, they interact with the lens system of the microscope, represented by a convolution with a Point Spread Function (PSF) kernel h_i. Additionally, we account for small translations $t_i \in \mathbb{R}^2$ by convolving with the translation kernel T_{t_i} and add a Gaussian noise variable ϵ_i to model various noise sources. The resulting image formation model is:

$$I_i = h_i * T_{t_i} * P_{R_i}[V] + \epsilon_i, \tag{2}$$

We assume square integrable functions, i.e. $P_{R_i}[V]$ and $I_i \in L^2(\mathbb{R}^2)$. The image formation model is usually formulated and implemented in Fourier space, which we describe in Apx. A.

3 RELATED WORK

Amortized inference of poses for cryo-EM Recently, deep learning based methods have been applied to the cryo-EM reconstruction problem (Donnat et al., 2022; Giri et al., 2023). With this, a new approach has emerged: amortized inference over poses. Such methods learn a parametrized function that estimates a pose given an image, as opposed to estimating the pose for each image individually. Typically, (variational) autoencoder architectures are employed in this setting. For example, Rosenbaum et al. (2021) use a variational autoencoder architecture in a heterogeneous setting in which the encoder predicts both poses and conformations. CryoVAEGAN (Miolane et al., 2020) is a pipeline in which an encoder estimates rotations as well as parameters of the contrast transfer function (CTF). In the spatialVAE pipeline the encoder estimates translations and in-plane rotations Bepler et al. (2019). CryoPoseNet (Nashed et al., 2021), cryoAI (Levy et al., 2022a) and cryoFIRE (Levy et al., 2022b) demonstrate the efficacy of an autoencoder (AE) architecture in which the encoder learns to estimate poses. While promising, amortized inference methods may suffer from convergence issues. For example, in (Rosenbaum et al., 2021), a strong prior on the backbone of the molecule is needed for the pipeline to converge. CryoPoseNet tends to get stuck in local minima while cryoAI and cryoFIRE limit this issue by using an additional symmetric loss.

Homoemorphic Encoders and Equivariance Falorsi et al. (2018) introduced *homeomorphic encoders* to describe a setting where the input data has a geometric structure associated with a symmetry group and the goal for the encoder is to learn a homeomorphic geometric representation of the data in the latent space. This is analogous the amortized inference of poses in cryo-EM, as each image is identified by an element of $SO(3)$. Esmaeili et al. (2023) show that these models are prone to optimization challenges due to topological defects at initialization that cannot be resolved continuously, partially explaining the convergence issues observed in previous works employing amortized inference of poses. One can incorporate geometric inductive bias into neural networks via

equivariance. For example, convolutional neural networks (CNNs) are translation-equivariant. This idea is generalized via group convolution (GCNNs) (Cohen & Welling, 2016a) and Steerable CNNs (Cohen & Welling, 2016b) to achieve equivariance with respect to more general symmetry groups. Cryo-EM images possess a number of symmetries (Cesa et al., 2022a) which can be exploited via an equivariant design. Previously, Nasiri & Bepler (2022); Cesa et al. (2023); Granberry et al. (2023) leveraged some of these planar symmetries. To the best of our knowledge, this is the first work to successfully leverage these symmetries in an end-to-end cryo-EM reconstruction pipeline.

4 METHOD

4.1 EQUIVARIANT AMORTIZED INFERENCE OF POSES

We can think of a 3D rotation as an element $R_i = (x_i, y_i, z_i) \in SO(3)$, where x_i, y_i and $z_i \in \mathbb{R}^3$ are the columns of the matrix R_i associated to the rotation. The vectors x_i, y_i and z_i form an orthonormal basis of $\mathbb{R}^3$. The operator P_{R_i} defined in Eq. 1 first rotates the volume V and then performs a projection along the z-axis. Rotating the volume V by R_i can be seen as transforming V with the transformation that would align x_i, y_i and z_i with the standard basis vectors of $\mathbb{R}^3$. Therefore, alternatively, we can see the axis z_i as the 'direction' along which the projection is performed. We shall call the axis z_i the *viewing direction*. Note that two images $I_i, I_j \in L^2(\mathbb{R}^2)$ that are generated by projecting along the same viewing direction z can only differ by an in-plane rotation $r_\theta \in \mathrm{SO}(2)$, whereas if they differ by a rotation and a reflection $f \cdot r_\theta \in \mathrm{O}(2)$, they are obtained by projecting along opposite viewing directions z and $-z$, see Fig. 1. In fact, given an element

$$g = f^c \cdot r_\theta = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^c \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \in \mathrm{O}(2) \tag{3}$$

with $c \in \{0, 1\}$ and $\theta \in [0, 2\pi)$, if two images I_i and I_j are related such that[1] $I_j = L_g[I_i]$, then we know that their associated poses $R_i, R_j \in \mathrm{SO}(3)$ are related by the following identity:

$$R_j = \varphi(g) R_i = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}^c \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} R_i. \tag{4}$$

where $\varphi : \mathrm{O}(2) \to \mathbb{R}^{3\times 3}$ is a three-dimensional representation of $\mathrm{O}(2)$; see Apx. D. Poses typically include also planar translations in $\mathbb{R}^2$, which follow an analogous equivariance rule; see Apx. E.1 for more details. In Sec. 4.3, we will exploit this knowledge in the amortized inference framework by restricting the parameterized function mapping images to their associated poses to be *equivariant* to a subgroup $H \subseteq \mathrm{O}(2)$.

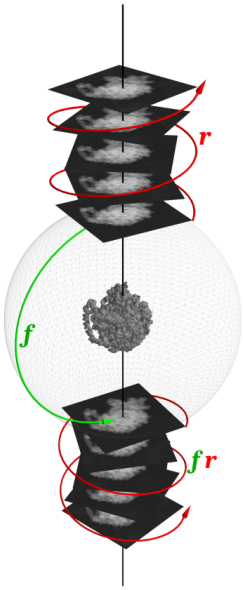


Figure 1: Images in the same equivalence class differ only by an in-plane rotation and/or reflection. Source: Cesa et al. (2022a).

4.2 PIPELINE OVERVIEW

We build on the work by Levy et al. (2022a), adapting their autoencoder pipeline by incorporating an equivariant encoder. Given an input image Y, the encoder $E\colon L^2(\mathbb{R}^2) \to \mathbb{R}^2 \times \mathrm{SO}(3)$ predicts a pose existing of a translation $t \in \mathbb{R}^2$ and a rotation $R \in \mathrm{SO}(3)$. The decoder maintains an estimate of the reconstructed volume via an implicit neural representation $\hat{V}\colon \mathbb{R}^3 \to \mathbb{C}$, which models the Fourier transform of V. The decoder simulates the image formation model in Fourier space by leveraging the *Fourier Slice Theorem* as in Apx. A. : it rotates a slice of three-dimensional coordinates $[k_x, k_y, 0]^\top$ by the rotation R and, then, feeds them into $\hat{V}$. The result is translated by t, convoluted with the contrast transfer function; see Apx. B for more details about the decoder. Finally, the output X is compared to the input Y via a mean-squared error loss ($L2$ loss).

4.3 EQUIVARIANT ENCODER

We replicate the architecture of the original encoder of cryoAI (see Apx. B of Levy et al. (2022a)) using the `escnn` Python library Cesa et al. (2022b) to incorporate equivariance to a subgroup $H \subset$

[1] The *left action* of $g \in \mathrm{O}(2)$ on an image $f \in L^2(\mathbb{R}^2)$ yields a new image given by $[L_g[f]](x) = f(g^{-1}x)$.

$O(2)$; see Apx. E for more details. We can interpret the encoder E as two separate functions: $E_r : L^2(\mathbb{R}^2) \to \mathrm{SO}(3)$ that outputs a rotation, and $E_t : L^2(\mathbb{R}^2) \to \mathbb{R}^2$ that outputs a translation. Equivariance ensures that, for any element $g \in H$ and image I:

$$E_r(L_g[I]) = \varphi(g)E_r(I), \quad \text{and} \quad E_t(L_g[I]) = g \cdot E_t(I). \tag{5}$$

In particular, we experimented with equivariance to the $H = C_4$ and $H = D_4$ subgroups of $O(2)$. C_4 contains the rotations by $0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$, while D_4 contains these rotations, and their combinations with a reflection. Note that these groups model the exact rotational symmetries of a pixel grid.

4.4 Symmetric loss function

The original pipeline is optimized using a *symmetric loss function* to reduce convergence issues Levy et al. (2022a). This loss function augments a batch with a duplicate of each image rotated by π. For each image and its rotated duplicate, the model is only supervised on the one it reconstructs best. Formally, the symmetric loss $\mathcal{L}_{\text{sym}}$ is defined as follows:

$$\mathcal{L}_{\text{sym}} = \sum_{i \in B} \min\{||\hat{Y}_i - \Gamma(Y_i)||^2, ||L_\pi[\hat{Y}_i] - \Gamma(L_\pi[Y_i])||^2\}, \tag{6}$$

where Γ is the whole cryoAI pipeline, interpreted as a function and L_π is the left action[1] of the rotation by π. In the equivariant pipeline, the loss function in Eq. 6 becomes trivial, as both C_4 and D_4 include the rotation by π, and thus $||\hat{Y}_i - \Gamma(Y_i)|| = ||R_\pi[\hat{Y}_i] - \Gamma(R_\pi[Y_i])||$. When we use $H = C_4$, which includes rotations but no reflections, we adapt the symmetric loss above by replacing L_π by L_f, where f is the reflection in Eq. 3. If L_f is used, we refer to the loss as 'mirror loss' $\mathcal{L}_{\text{mir}}$.

5 Experiments

Simulated datasets with varying noise levels were generated as in Levy et al. (2022a) from the following Protein Data Bank structures: plasmodium falciparum 80S ribosome (PDB: 3J79 and 3J7A) Wong et al. (2014) (`80S`), and the pre-catalytic spliceosome (PDB: 5NRL) Plaschka et al. (2017) (`spliceosome`). See Apx. C for more details.

5.1 Convergence of original pipeline

To examine the convergence progress of the original pipeline, cryoAI was trained with a non-equivariant encoder and $\mathcal{L}_{\text{sym}}$ across datasets for two structures - each with a noiseless and an SNR $= 1$ version - using 5 different seeds for 6000 seconds. Fig. 2 illustrates the trends of Mean Squared Error (MSE) and Median Squared Error (MedSE) for 5 runs using a noiseless `spliceosome` dataset. The observed pattern shows an initial decline in both MedSE and MSE, with MedSE approaching zero fast and MSE plateauing, then sharply decreasing. This behavior, further analyzed in Apx. G, correlates with transient spurious planar symmetries in volume reconstructions, which resolve as MSE improves. While $\mathcal{L}_{\text{sym}}$ reduces the duration of such states, they still occur.

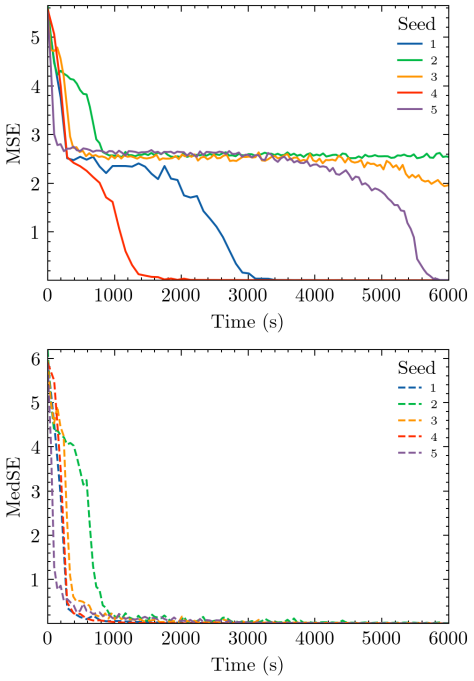


Figure 2: The MSE and MedSE over time for each of the five runs on the *noiseless* `spliceosome` dataset.

5.2 Impact of equivariance on convergence of pipeline

We evaluated $H = C_4$- and D_4-equivariant encoders against a standard model using $\mathcal{L}_{\text{mir}}$ or an $L2$ loss across four simulated datasets (each for two structures, with and without noise), with 8 seeds per dataset and a 6000-second training limit. Performance metrics including MSE, convergence rates,

and volume resolution are summarized in Tab. 1, with metric definitions in Apx. F. Convergence is defined as MSE < 0.1 based on experiments with the standard pipeline that showed that the volume escapes symmetric states after the MSE has decreased sufficiently. Our findings reveal that C_4 equivariance with $L2$ loss rarely converges, but it *significantly enhances convergence speed and accuracy when paired with* $\mathcal{L}_{\text{mir}}$. D_4-equivariant encoders *reliably converge without* $\mathcal{L}_{\text{mir}}$, outperforming all other models in speed of convergence and metric performance, as highlighted by the superior resolution and pose estimation accuracy. Moreover, we compare both the Fourier Shell Correlation (FSC) curves and the 3D reconstructions of some of the models in Fig. 3. The D_4-equivariant models *consistently yield higher FSC curves, indicating improved reconstruction quality*. However, we note that the improved reconstruction quality might still not be enough to provide additional biological insights. Since we are not sufficiently expert to judge this aspect, we leave it for future exploration but emphasize the computational benefits of our method with respect to the standard cryoAI pipeline, particularly in terms of faster reconstruction.

Table 1: Average values for the MSE, MedSE, convergence time in seconds, number of iterations until convergence, resolution in Angstrom and the percentage of runs that converged. The average is taken over the three best runs in terms of final MSE. The resolution is reported using the FSC $= 0.5$ criterion. For each metric, the best value is indicated in bold. A dash (-) indicates that at least one of the three best runs did not converge.

Dataset	Model	MSE	MedSE	Conv. time (s)	Steps until conv.	Resolution (Å)	Conv. Runs %
	Standard ($\mathcal{L}_{\text{sym}}$)	0.0040 ± 0.0002	0.0030 ± 0.0001	2818 ± 286	5800 ± 589	11.26 ± 0.11	62.5
`Spliceosome`	C_4 (L2)	2.5893 ± 0.0322	0.0166 ± 0.0090	-	-	12.18 ± 0.13	0
noiseless	C_4 ($\mathcal{L}_{\text{mir}}$)	0.0008 ± 0.0001	0.0005 ± 0.0000	1457 ± 1001	**1967 ± 1347**	11.11 ± 0.19	100
	D_4 (L2)	**0.0004 ± 0.0000**	**0.0003 ± 0.0000**	**1239 ± 782**	2800 ± 1766	**10.88 ± 0.18**	87.5
	Standard ($\mathcal{L}_{\text{sym}}$)	0.0041 ± 0.0001	0.0030 ± 0.0001	2564 ± 394	5300 ± 804	10.20 ± 0.10	100
`80S`	C_4 (L2)	0.8160 ± 1.1520	0.0020 ± 0.0016	-	-	10.35 ± 0.28	25
noiseless	C_4 ($\mathcal{L}_{\text{mir}}$)	0.0010 ± 0.0000	0.0006 ± 0.0000	791 ± 195	**1067 ± 262**	10.05 ± 0.00	100
	D_4 (L2)	**0.0005 ± 0.0001**	**0.0004 ± 0.0000**	**774 ± 143**	1767 ± 330	**9.72 ± 0.09**	87.5
	Standard ($\mathcal{L}_{\text{sym}}$)	0.3232 ± 0.3384	0.0077 ± 0.0031	-	-	12.72 ± 0.97	12.5
`Spliceosome`	C_4 (L2)	2.6525 ± 0.0104	0.0308 ± 0.0273	-	-	13.06 ± 0.15	0
SNR 1	C_4 ($\mathcal{L}_{\text{mir}}$)	0.0014 ± 0.0001	0.0010 ± 0.0001	**1181 ± 312**	**1600 ± 432**	11.33 ± 0.00	87.5
	D_4 (L2)	**0.0010 ± 0.0000**	**0.0007 ± 0.0000**	1624 ± 426	3333 ± 998	**11.03 ± 0.10**	50
	Standard ($\mathcal{L}_{\text{sym}}$)	0.0102 ± 0.0003	0.0070 ± 0.0009	3093 ± 418	6400 ± 909	11.22 ± 0.00	75
`80S`	C_4 (L2)	0.0055 ± 0.0014	0.0037 ± 0.0009	4080 ± 999	10100 ± 2491	10.97 ± 0.20	37.5
SNR 1	C_4 ($\mathcal{L}_{\text{mir}}$)	0.0038 ± 0.0002	0.0027 ± 0.0001	1386 ± 808	**1867 ± 1087**	10.97 ± 0.00	100
	D_4 (L2)	**0.0032 ± 0.0001**	**0.0023 ± 0.0000**	**1071 ± 135**	2433 ± 309	**10.65 ± 0.22**	100

Higher noise levels To investigate if the convergence benefits of equivariance are maintained under higher noise conditions, we conducted an experiment focusing on the standard model and the D_4-equivariant model. These models were trained on datasets generated using the `80S` ribosome structure at various noise levels; see Apx. C. Each model is trained for 6 hours. We recorded the metrics for the best run out of 5 for each model and dataset. The results in Tab. 2 show that the benefits of equivariance hold at higher noise levels. The equivariant encoder outperforms the standard encoder on all metrics for each noise level, except the highest (SNR $= 0.25$), although it should be noted that all seeds of both models didn't converge at that noise level.

6 Final Discussions

Our experiments demonstrate the benefits of incorporating geometric priors in the Cryo-EM amortized inference framework. Indeed, equivariant models consistently show drastic improvements in convergence speed, convergence ratio and final resolution. In particular, D_4-equivariance makes the expensive symmetric loss superfluous. These results indicate that apart from the symmetric loss, equivariant amortized inference can be a viable alternative solution to the convergence issues associated with symmetric reconstructions. Still, some runs with equivariant encoder did not convergence; in this context, Esmaeili et al. (2023) suggested that, while inductive biases like equivariance can guide models towards learning homeomorphic representations, they might also complicate the training process in practice. Inspired by their solution, we find encoders modeling a multimodal distribution, as opposed to our single-pose encoders, to be a promising direction of research.

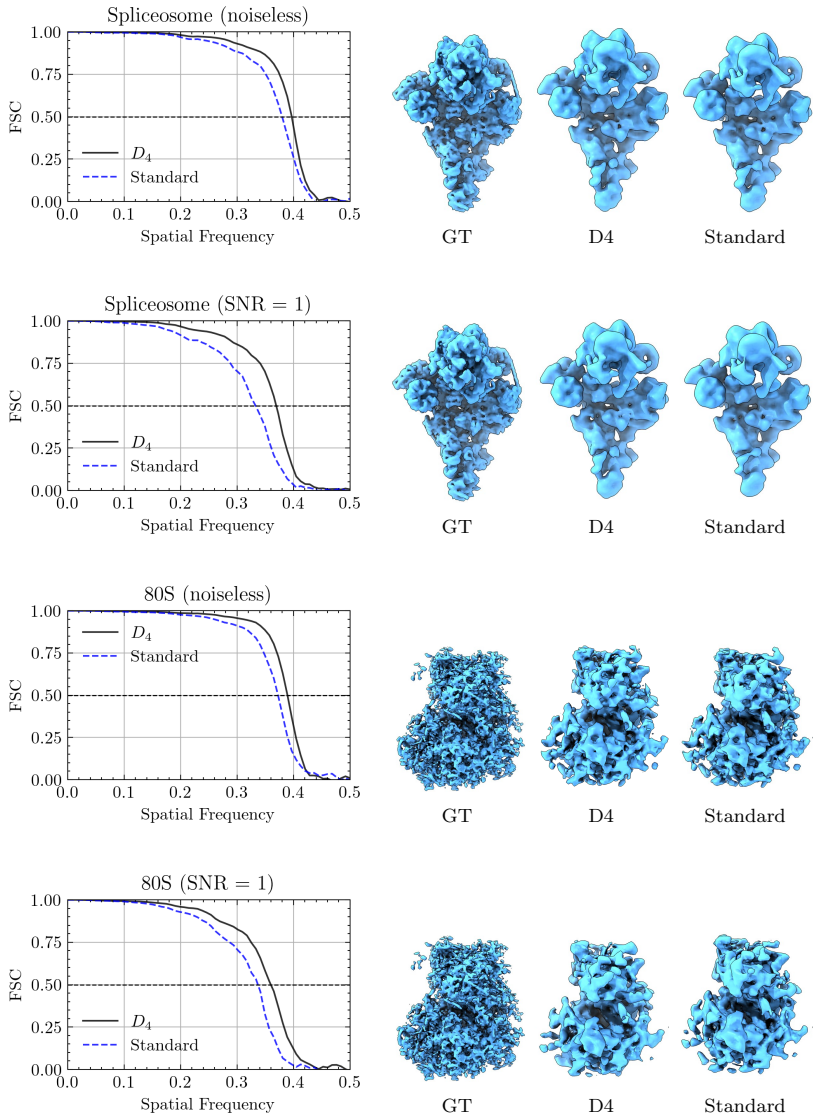


Figure 3: Fourier Shell Correlation (FSC) curves and reconstruction visualizations comparing the reconstruction to the ground truth (GT) for the top-performing seeds in 6000-second runs across four simulated datasets. Each dataset is represented by two curves and two visualizations: one from the best run using the D_4-equivariant model and one from the best run with the standard cryoAI pipeline. The volume visualisations were made with software ChimeraX (Meng et al. (2023)). The D_4-equivariant model consistently improves the FSC score over the non-equivariant cryoAI baseline.

Table 2: Pose estimation error, convergence time, and final resolution of the best model across 5 seeds for both model types for each of the noise levels. Resolution is reported using the FSC $= 0.5$ criterion.

SNR	Model	MSE	MedSE	Convergence time (s)	Final resolution (Å)
0.75	D_4 (L2)	**0.0012**	**0.0009**	**1238.04**	**9.46**
	Standard ($\mathcal{L}_{\text{sym}}$)	0.0027	0.0021	3065.58	9.65
0.625	D_4 (L2)	**0.0013**	**0.0011**	**1136.72**	**9.56**
	Standard ($\mathcal{L}_{\text{sym}}$)	0.0036	0.0025	2722.84	9.85
0.5	D_4 (L2)	**0.0021**	**0.0014**	**1850.60**	**9.75**
	Standard ($\mathcal{L}_{\text{sym}}$)	0.0057	0.0039	3885.72	10.72
0.375	D_4 (L2)	**0.0057**	**0.0033**	**6263.10**	**10.97**
	Standard ($\mathcal{L}_{\text{sym}}$)	0.0780	0.0096	12741.46	12.37
0.25	D_4 (L2)	1.4671	0.1627	-	22.98
	Standard ($\mathcal{L}_{\text{sym}}$)	**0.7455**	**0.0546**	-	**16.09**

REFERENCES

Tristan Bepler, Ellen Zhong, Kotaro Kelley, Edward Brignole, and Bonnie Berger. Explicitly disentangling image content from translation and rotation with spatial-vae. *Advances in Neural Information Processing Systems*, 32, 2019.

Gabriele Cesa, Arash Behboodi, Taco S Cohen, and Max Welling. On the symmetries of the synchronization problem in cryo-em: Multi-frequency vector diffusion maps on the projective plane. *Advances in Neural Information Processing Systems*, 35:5446–5458, 2022a.

Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build E(N)-equivariant steerable CNNs. In *International Conference on Learning Representations*, 2022b. URL `https://openreview.net/forum?id=WE4qe9xlnQw`.

Gabriele Cesa, Kumar Pratik, and Arash Behboodi. Equivariant self-supervised deep pose estimation for Cryo EM. In *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, Proceedings of Machine Learning Research. PMLR, 28 Jul 2023.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016a.

Taco S Cohen and Max Welling. Steerable CNNs. *arXiv preprint arXiv:1612.08498*, 2016b.

Claire Donnat, Axel Levy, Frederic Poitevin, Ellen D Zhong, and Nina Miolane. Deep generative modeling for volume reconstruction in cryo-electron microscopy. *Journal of Structural Biology*, pp. 107920, 2022.

Babak Esmaeili, Robin Walters, Heiko Zimmermann, and Jan-Willem van de Meent. Topological obstructions and how to avoid them. *arXiv preprint arXiv:2312.07529*, 2023.

Luca Falorsi, Pim De Haan, Tim R Davidson, Nicola De Cao, Maurice Weiler, Patrick Forré, and Taco S Cohen. Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*, 2018.

Nabin Giri, Raj S Roy, and Jianlin Cheng. Deep learning for reconstructing protein structures from cryo-em density maps: Recent advances and future directions. *Current Opinion in Structural Biology*, 79:102536, 2023.

Darnell Granberry, Simons Machine Learning Center, Alireza Nasiri, Jiayi Shou, Alex J Noble, and Tristan Bepler. So (3)-equivariant representation learning in 2d images. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023.

Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. *Advances in Neural Information Processing Systems*, 33:22554–22565, 2020.

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. In *European Conference on Computer Vision*, pp. 540–557. Springer, 2022a.

Axel Levy, Gordon Wetzstein, Julien NP Martel, Frederic Poitevin, and Ellen Zhong. Amortized inference for heterogeneous reconstruction in cryo-em. *Advances in Neural Information Processing Systems*, 35:13038–13049, 2022b.

Elaine C Meng, Thomas D Goddard, Eric F Pettersen, Greg S Couch, Zach J Pearson, John H Morris, and Thomas E Ferrin. Ucsf chimerax: Tools for structure building and analysis. *Protein Science*, 32(11):e4792, 2023.

Nina Miolane, Frédéric Poitevin, Yee-Ting Li, and Susan Holmes. Estimation of orientation and camera parameters from cryo-electron microscopy images with variational autoencoders and generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 970–971, 2020.

Youssef SG Nashed, Frédéric 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4066–4076, 2021.

Alireza Nasiri and Tristan Bepler. Unsupervised object representation learning using translation and rotation group equivariant vae. *Advances in Neural Information Processing Systems*, 35:15255–15267, 2022.

Clemens Plaschka, Pei-Chun Lin, and Kiyoshi Nagai. Structure of a pre-catalytic spliceosome. *Nature*, 546(7660):617–621, 2017.

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, 2017.

Dan Rosenbaum, Marta Garnelo, Michal Zielinski, Charlie Beattie, Ellen Clancy, Andrea Huber, Pushmeet Kohli, Andrew W Senior, John Jumper, Carl Doersch, et al. Inferring a continuous distribution of atom coordinates from cryo-em images using vaes. *arXiv preprint arXiv:2106.14108*, 2021.

Sjors HW Scheres. Relion: implementation of a bayesian approach to cryo-em structure determination. *Journal of structural biology*, 180(3):519–530, 2012.

Amit Singer and Yoel Shkolnisky. Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming. *SIAM journal on imaging sciences*, 4 (2):543–572, 2011.

Amit Singer and Fred J Sigworth. Computational methods for single-particle electron cryomicroscopy. *Annual review of biomedical data science*, 3:163–190, 2020.

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.

Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. *Advances in neural information processing systems*, 32, 2019.

Wilson Wong, Xiao-chen Bai, Alan Brown, Israel S Fernandez, Eric Hanssen, Melanie Condron, Yan Hong Tan, Jake Baum, and Sjors HW Scheres. Cryo-em structure of the plasmodium falciparum 80s ribosome bound to the anti-protozoan drug emetine. *Elife*, 3:e03080, 2014.

A IMAGE FORMATION MODEL IN FOURIER SPACE

Implementing the image formation model in reconstruction algorithms can become computationally expensive due to the rotation of the volume V and the integral in equation 1. Formulating the image formation in Fourier space allows for the application of the *Fourier Slice Theorem*, which makes rotating and integrating the volume redundant.

The theorem states that evaluating a volume on a plane ('a slice of coordinates') in Fourier space gives the Fourier transform of a projection of that volume along an axis perpendicular to that plane. In mathematical terms, this means that

$$S_R[\mathcal{F}_3(V)] = \mathcal{F}_2(P_R[V]) \tag{7}$$

where $\mathcal{F}_2$ and $\mathcal{F}_3$ are the two- and three dimensional Fourier transforms, and S_R is an operator that performs the operation of evaluating a volume on a plane, and is defined such that

$$S_R[V]\colon (k_1, k_2) \to V(R^\top \cdot (k_1, k_2, 0)^\top). \tag{8}$$

Using the *Fourier Slice Theorem*, we can formulate the entire image formation model in Fourier space as follows:

$$\hat{I}_i = \hat{h}_i \odot \hat{T}_{t_i} \odot S_{R_i}[\mathcal{F}_3(V)] + \hat{\epsilon}_i \tag{9}$$

Here $\hat{h}_i$ is the Fourier transform of the h_i kernel, called the Contrast Transfer Function (CTF), $\hat{T}_{t_i}$ is the translation operator performing t_i in Fourier space (which corresponds to a phase shift) and $\hat{\epsilon}_i$ is i.i.d. complex Gaussian noise at each frequency. The symbol $\odot$ stands for element-wise multiplication that replaces convolution in the real formulation, which follows from the Convolution Theorem.

B DECODER ARCHITECTURE

The decoder takes the rotation R_i and the translation t_i outputted by the encoder as an input. An array of L^2 three-dimensional coordinates $[k_x, k_y, 0]^\top \in \mathbb{R}^3$ is generated that represent the Fourier frequencies corresponding to a grid of L^2 coordinates on the $xy-$plane. Each coordinate in this slice is then rotated by $R_i^\top$ and fed into a neural network $\hat{V}\colon \mathbb{R}^3 \to \mathbb{C}$. The network $\hat{V}$, called FourierNet, represents the Fourier transform of the electrostatic potential of the molecule that we wish to reconstruct.

FourierNet consists of two SIREN models that take a coordinate $[k_x, k_y, k_z] \in \mathbb{R}^3$ and output a vector in $\mathbb{R}^2$. SIREN (Sinusoidal Representation) networks are fully connected feed-forward neural networks that utilize sinusoidal activation functions. SIREN networks have been shown to be successful in representing complex signals defined on real space. (Sitzmann et al. (2020)) The Fourier transform of an electrostatic of a molecule is known to vary in magnitude over several orders of magnitude. (Levy et al. (2022a)) To allow FourierNet to represent a function that varies several orders of magnitude in value, one of the two SIRENs has the property that the exponential function is applied element-wise on its output.

To ensure that the network $\hat{V}$ actually represents a Fourier transform, the following property is enforced

$$\hat{V}(k) = \hat{V}(-k)^* \qquad \text{if } k_x < 0 \tag{10}$$

simply by following this as a definition. The two output vectors of the networks are multiplied element-wise and the result is interpreted as an element of $\mathbb{C}$ via the mapping

$$[v_1, v_2]^\top \mapsto v_1 + v_2 i.$$

Since the input to FourierNet was a slice of three-dimensional coordinates, the output represents a slice of a Fourier transform $S_{R_i}[\mathcal{F}_3(V)] \in \mathbb{C}^{L\times L}$ as defined in Eq. 7. The final output X_i of the decoder is obtained as follows:

$$\hat{X}_i = \hat{h}_i \odot \hat{T}_{t_i} \odot S_{R_i}[\mathcal{F}_3(V)], \tag{11}$$

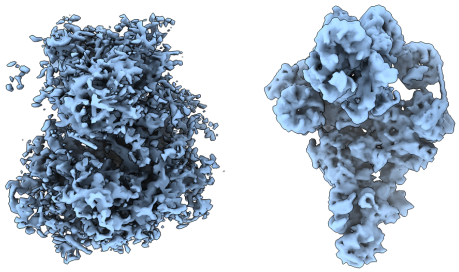
where $\hat{h}_i$ is the CTF, and $\hat{T}_{t_i}$ is the translation operator performing the translation t_i predicted by the encoder in Fourier space. The CTF is a function of parameters that can be estimated from the dataset and are assumed to be known. For more details about the pipeline architecture, see Levy et al. (2022a).

C DATA

The data was synthesized with structures from the Protein Data Bank. The two structures that were used were the plasmodium falciparum 80S ribosome structure (PDB: 3J79 and 3J7A) Wong et al. (2014) and the pre-catalytic spliceosome structure (PDB: 5NRL) Plaschka et al. (2017).

The datasets were generated by simulating the image formation model described in Sec. 2 using ground-truth volume represented as a three-dimensional voxel grid, as in Levy et al. (2022a). The volume data for the ground truth volumes was generated in ChimeraX Meng et al. (2023) as described in Apx. A.1 of Levy et al. (2022a). See Fig. 4 for a visualization of the volumes.

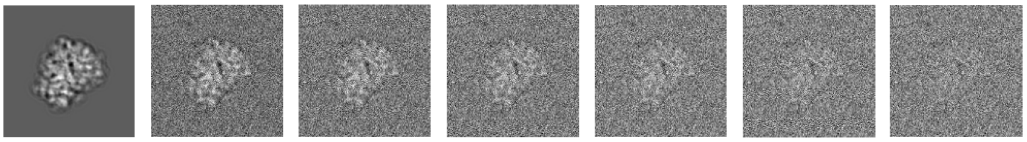
Each dataset contains $100\,000$ images of dimension 128×128. The simulator uses poses $R_i \in \mathrm{SO}(3)$ randomly sampled from a uniform distribution over $\mathrm{SO}(3)$, and translations $t_i \in \mathbb{R}^2$ sampled from a Gaussian distribution with mean zero and a standard deviation of 20Å.



The plasmodium falciparum 80S ribosome volume. The pre-catalytic spliceosome volume.

Figure 4: Visualization of the two volumes that were used to generate the datasets. The visualisations were made with software ChimeraX Meng et al. (2023)

For both the pre-catalytic spliceosome and the plasmodium falciparum 80S ribosome structures, one noisless dataset was generated, and one with signal-to-noise ratio (SNR) equal to 1. For the plasmodium falciparum 80S ribosome structure, additional datasets were generated with SNR $0.75, 0.625, 0.5, 0.375$ and 0.25, in Fig. 5 we show examples of images from each of the datasets generated with the plasmodium falciparum 80S ribosome.



Noiseless SNR 1 SNR 0.75 SNR 0.625 SNR 0.5 SNR 0.375 SNR 0.25

Figure 5: Examples of images from our datasets generated with the plasmodium falciparum 80S ribosome volume with different noise levels.

D BACKGROUND ON GROUP AND GROUP REPRESENTATION THEORY

Group theory, and specifically group representation theory, are important building blocks in the theory about equivariant neural networks, such as steerable networks that are applied in this work. In this Section we provide some important definitions and preliminaries that are useful for understanding the method presented in this work.

Groups can be useful to describe the structure of sets of geometrical transformations, such as rotations and translations. We start by giving the definition of a group.

Definition A *group* is a set G equipped with a group product $\cdot$ satisfying the following properties:

- Closure: If $g, h \in G$, then also $g \cdot h \in G$.
- Associativity: For $g, h, i \in G$, $(g \cdot h) \cdot i = g \cdot (h \cdot i)$.
- Identity: There exists an element $e \in G$ such that $e \cdot g = g \cdot e = g$ for all $g \in G$.
- Inverse element: for all $g \in G$, there exists an element $g^{-1} \in G$ such that $g \cdot g^{-1} = g^{-1} \cdot g = e$.

Example The *circle group* is the set of all elements of $\mathbb{C}$ with absolute value equal to one, denoted

$$S^1 = \{z \in \mathbb{C} \,|\, |z| = 1\}$$

with as group product multiplication of complex numbers and as identity element 1.

Another important definition in group theory is that of a *group homomorphism*. Intuitively, group homomorphisms are functions between groups that preserve algebraic structure shared by these groups.

Definition If G_1 and G_2 are groups, then a function $f\colon G_1 \to G_2$ is a group homomorphism if for all $g, h \in G_1$

$$f(g \cdot h) = f(g) \star f(h).$$

Here $\cdot$ is used to denote the group product of G_1 and $\star$ for the group product of G_2.

If $G_1 = G_2$, f is called an automorphism. If f is bijective, it is called a *group isomorphism*. If there is a group isomorphism between G_1 and G_2, we call G_1 and G_2 isomorphic and we write $G_1 \cong G_2$.

The set of all automorphisms of a group G is again a group with map composition as a group product. It is denoted by $\mathrm{Aut}(G)$.

It can be useful to 'combine' several groups into one. In this way we can represent different types of transformations (e.g. translation and rotation) as elements of a single group. One way to do this is via *semi-direct products*.

Definition Let H and N be groups. Let $\tau : H \to \mathrm{Aut}(N)$ be a group homomorphism. We define the semi-direct product of H with N with respect to τ, written $N \rtimes_\tau H$, as the set $N \times H$ with the following group operation

$$(n_1, h_1) \cdot (n_2, h_2) = (n_1 \tau(h_1)(n_2), h_1 h_2).$$

As mentioned, some groups describe the structure of sets of geometrical transformations, such as rotations and translations. Such groups have *actions* on other spaces. Group representations are one way to describe the actions of certain groups (linear transformations) on vector spaces. To define group representations, we first give the definition of the general linear group:

Definition The general linear group of a vector space V, denoted $\mathrm{GL}(V)$, is the set of all invertible linear maps from V to V with as a group product composition of maps.

Since in deep learning we are often working with real-valued vectors, when we refer to the general linear group, we are usually talking about $\mathrm{GL}(\mathbb{R}^n)$. This is the group of all invertible $n \times n$ matrices.

Using the definition of the general linear group, we can define group representations.

Definition A *representation* of a group G is a homomorphism $\varphi\colon G \to \mathrm{GL}(V)$ for some vector space V. We can write $\varphi(g)$ or φ_g for φ evaluated at g.
We write d_φ to denote the *dimension* of the representation, which is equal to the dimension of the vector space V.

Some examples of representations follow below.

Example For any group G we can define the representation $\varphi\colon G \to \mathrm{GL}(\mathbb{R})$ as $\varphi(g) = 1$ for all $g \in G$. This is called the *trivial representation*.

By looking at a representation of the circle group S^1 that we defined before, we see that the circle group is actually also a group that describes a specific type of transformation on $\mathbb{R}^2$: rotations.

Example We can define a representation $\varphi\colon S^1 \to \mathrm{GL}(\mathbb{R}^2)$ as follows:

$$\varphi(e^{i\theta}) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

Every element of S^1 is mapped onto a rotation matrix by this representation, which shows us that S^1 has an action on $\mathbb{R}^2$ corresponding to planar rotation around the origin.

The group formed by all 2×2 rotation matrices is called $\mathrm{SO}(2)$. One can verify that φ is a isomorphism between S^1 and $\mathrm{SO}(2)$, which means that $S^1 \cong \mathrm{SO}(2)$.

We can combine multiple representations together into a single representation via the *direct sum*. This concept is important in the theory of steerable CNNs.

Definition Let $\rho_1\colon G \to \mathrm{GL}(V_1)$ and $\rho_2\colon G \to \mathrm{GL}(V_2)$ be two representations of G. Then $(\rho_1 \oplus \rho_2)\colon G \to \mathrm{GL}(V_1 \oplus V_2)$, given by

$$(\rho_1 \oplus \rho_2)(g) = \begin{bmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{bmatrix} \tag{12}$$

is called their direct sum. The direct sum $(\rho_1 \oplus \rho_2)$ is again a representation. Since it is a diagonal matrix with ρ_1 and ρ_2 on the diagonals, it works on the vector space $(V_1 \oplus V_2)$ by letting ρ_1 work on the subspace V_1, and ρ_2 on the subspace V_2.

An important representation in the theory of steerable CNNs is the *regular representation*.

Example Let G be a finite group. If $f\colon G \to \mathbb{R}$ is a function over G and $g \in G$, then

$$[g.f](h) = f(g^{-1}h)$$

describes an action of G on the vector space of functions over G. The regular representation ρ_{reg} of G is the representation that describes this action.

The vector space of functions over G is equivalent to $\mathbb{R}^{|G|}$, since we can interpret a function $f\colon G \to \mathbb{R}$ as a vector where the ith element represents the function value assigned to the ith group element g_i. For an element $g \in G$ and $f \in \mathbb{R}^{|G|}$, ρ_{reg} associates to g the permutation matrix that sends element i to the element j such that $g^{-1} \cdot g_i = g_j$.

We finish this section with some important groups that we regularly refer to in this work. The first group, $\mathrm{SO}(3)$, contains all three-dimensional rotation matrices.

Example The *special orthogonal group* $\mathrm{SO}(3)$ consists of all 3×3 real orthogonal matrices R with a determinant of $+1$, representing three-dimensional rotation matrices. Specifically,

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times3} \mid R^T R = I \text{ and } \det(R) = 1\}.$$

Here, R^T denotes the transpose of R, I is the 3×3 identity matrix, and the group operation is matrix multiplication.

Another important group, $O(2)$, contains two-dimensional rotations and reflections.

Example The *orthogonal group* $O(2)$ consists of all 2×2 real orthogonal matrices A, which includes both rotations and reflections in two-dimensional space. It is defined as:

$$O(2) = \{A \in \mathbb{R}^{2\times2} \mid A^T A = I\},$$

where A^T is the transpose of A, and I is the 2×2 identity matrix. The group operation is matrix multiplication.

Finally, the groups C_4 and D_4 are both subgroups of $\mathrm{O}(2)$.

Example The *cyclic group* C_4 consists of the set of rotational symmetries of a square, containing four elements corresponding to rotations by multiples of 90 degrees. It is defined as:

$$C_4 = \{e, r, r^2, r^3\},$$

where e is the identity element (no rotation), and r is a 90-degree rotation. The group operation is the composition of rotations.

Example The *dihedral group* D_4, representing the symmetries of a square, consists of eight elements: four rotations and four reflections. It can be described as:

$$D_4 = \{r, s \mid r^4 = s^2 = 1,\ srs = r^{-1}\},$$

here r represents a 90-degree rotation, s represents a reflection, and r^{-1} is the inverse of r.

E EQUIVARIANT ENCODER ARCHITECTURE

A high-level overview of the equivariant encoder is shown in Figure 6. As mentioned in Section 4.2, the architecture was kept as similar as possible to the standard pipeline in Levy et al. (2022a). The encoders were implemented as steerable CNNs Cohen & Welling (2016b) with the `escnn` Python library Weiler & Cesa (2019); Cesa et al. (2022b)

If a symmetric- or mirror loss function (see Section 4.4) is used, the input Y_i is fed into a sym layer. This layer performs the operation of duplicating the batch of images and then rotating it by π or mirroring it with respect to the x-axis (depending on the type of loss). The transformed images are concatenated batch-wise. If an L2 loss function is used, this process is skipped. The batch then goes into a GaussianPyramid layer with five Gaussian filters of size 11 with cutoff frequencies that are geometrically distributed between 0.1 and 10 pix^{-1} (as in Levy et al. (2022a)), that filter each image.

The filtered images are fed into five consecutive EquivariantDoubleConvBlock networks. Each EquivariantDoubleConvBlock layer is made up of two convolutional layers with steerable kernels (R2Conv) and a PointwiseMaxPool layer. Each of the R2Conv layers is followed by a pointwise ReLU activation function and an InnerBatchNorm layer.

After passing through each of the EquivariantDoubleConvBlock blocks, the batch passes through a PointwiseMaxPool layer and a single R2Conv layer with a kernel of size 2. This results in an output with a width and height of 1, with overal shape (batch size) $\times$ (number of feature fields $\cdot\, |G|$) $\times$ (1) $\times$ (1). This output is fed into two separate equivariant fully connected networks (EquivariantFCBlock).

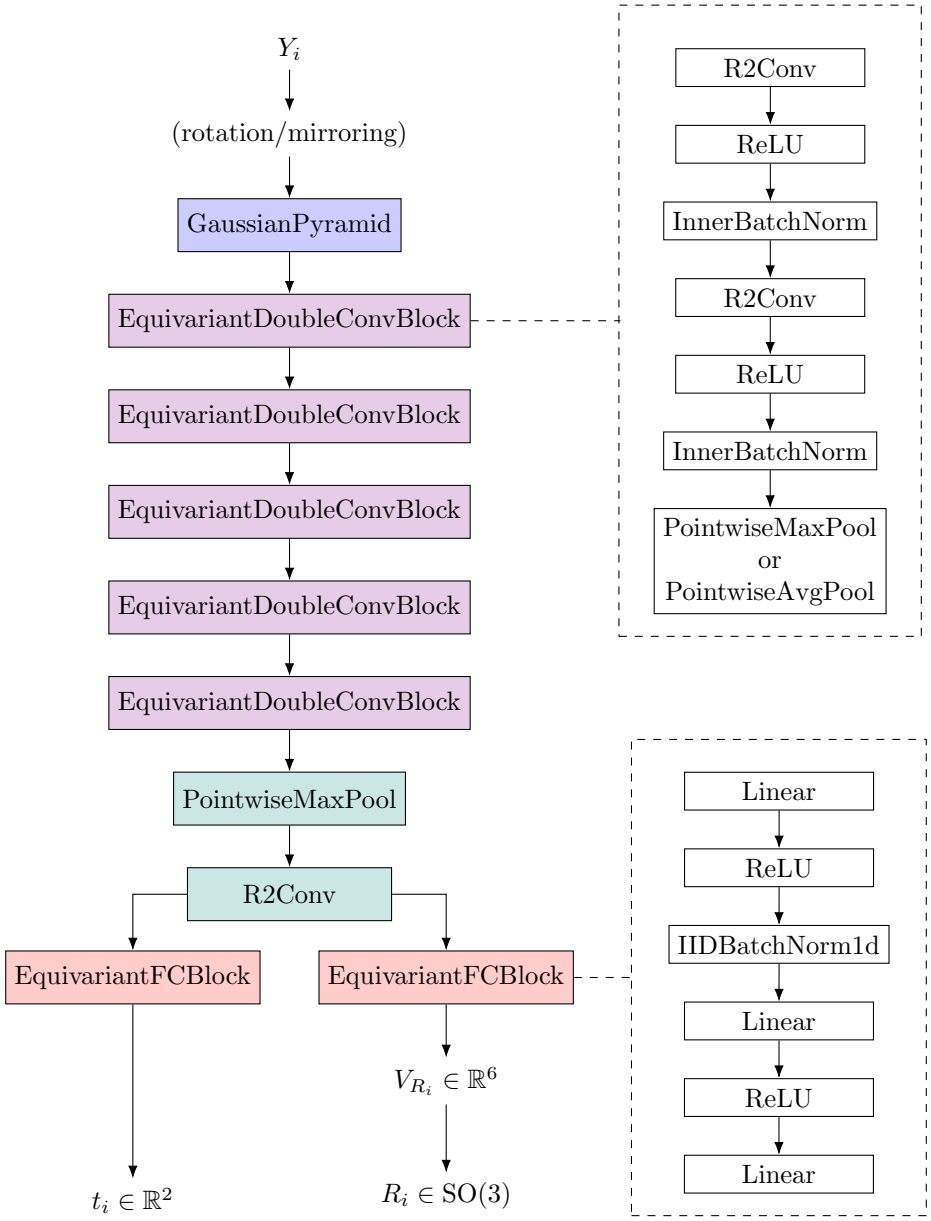


Figure 6: Visualization of our architecture of the equivariant encoder.

The first EquivariantFCBlock outputs a vector $t_i \in \mathbb{R}^2$, which is directly interpreted as a translation. The second EquivariantFCBlock outputs a vector $V_{R_i} \in \mathbb{R}^6$, which is then mapped to $R_i \in \mathrm{SO}(3)$ and interpreted as a rotation matrix as explained in Apx. E.1.

E.1 TECHNICAL DETAILS

In this section we present some technical details related to the implementation of the encoder. We start with some very brief theory.

Recall that in regular CNNs, layers are convolution operators that, when applied to an input, yield a *feature map*. A key difference between steerable CNNs and regular CNNs is that in the former, feature maps are replaced by steerable feature fields:

Definition A steerable feature field is a vector field $f : \mathbb{R}^2 \to \mathbb{R}^c$ defined on the base space $\mathbb{R}^2$, together with a transformation law that defines how the group $G = \mathbb{R}^2 \rtimes H$ acts on the field f as a whole.
The transformation law given an element $g = (t, h) \in G$ has the following form:

$$[g.f](x) = \rho(h)f(g^{-1}x). \tag{13}$$

where $\rho : H \to \mathrm{GL}(\mathbb{R}^c)$ is a representation of the group H. It defines how H acts on $\mathbb{R}^c$. The representation ρ is called the *type* of this particular feature field.

Layers in steerable CNNs are equivariant, meaning that if a feature field is transformed by a group element according to its transformation law and then passes through a layer, we want this to give the same output as the feature field first passing through that layer and then transforming the generated feature field by the same group elements according to its own transformation law. Mathematically, we can write this as:

$$(k \star t_l(g)[f]) = t_{l+1}(g)[(k \star f)] \tag{14}$$

for all $g \in G$. Here t_n is the transformation law of the feature field in layer n (defined as in equation 13), and $\star$ is the convolution operator.

Implementation As mentioned, the equivariant encoder was implemented using steerable CNNs. As part of the implementation of networks with the `escnn` library, one specifies both an *input type* and an *output type* for each layer, which together define the type of equivariance of the layer exhibits (as explained above).

The input type of the first DoubleConv block is a direct sum of five trivial representations, since its input is a stack of five filtered images that transform separately. The input type of each consecutive DoubleConv block is a direct sum of multiple regular representations. The number of regular representations in the direct sum is a hyperparameter that is different for each of the encoders. The features in each layer in the EquivariantFCBlock also transform according to a direct sum of multiple regular representations.

The output types of both EquivariantFCBlocks are especially important, since they determine the overall type of equivariance the encoder exhibits. Recall that we want the encoder to satisfy the following equations:

$$E_r(L_g[I]) = \varphi_r(g)E_r(I), \tag{15}$$

and

$$E_t(L_g[I]) = \varphi_t(g) \cdot E_t(I). \tag{16}$$

where $\varphi_t(g) : G \to \mathrm{GL}(\mathbb{R}^2)$ and $\varphi_r(g) : G \to \mathrm{GL}(\mathbb{R}^3)$ are two representations of G such that for $g = f^c \cdot r_\theta \in G$,

$$\varphi_t(g) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^c \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{17}$$

and

$$\varphi_r(g) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}^c \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

E_t satisfies equation 16 if the output type $\rho_{E_t}^{\text{out}}$ of the first EquivariantFCBlock is equal to φ_t, since E_t directly outputs t_i.

For the second EquivariantFCBlock we choose output type $\rho_{E_r}^{\text{out}} = \varphi_r \oplus \varphi_r$. Recall that the output of this EquivariantFCBlock is a vector $V_{R_i} \in \mathbb{R}^6$, which is then mapped to $R_i \in \text{SO}(3)$ and interpreted as a rotation matrix (see Figure 6). Therefore, we also have to define a mapping $V_{R_i} \mapsto R_i$ in such a way that equation 15 is satisfied. This is done as follows:

Let $V_{R_i} = [v_1 \ \dots \ v_6]^\top$, $V_1 = [v_1 \ v_2 \ v_3]^\top$, $V_2 = [v_4 \ v_5 \ v_6]^\top$ and $V_3 = V_1 \times V_2$, where $\times$ indicates the cross product[2]. We define

$$M = \begin{bmatrix} | & | & | \\ V_1 & V_2 & V_3 \\ | & | & | \end{bmatrix} \tag{19}$$

Let U_M, S_M, V_M be the Singular Value Decomposition (SVD) of M, and define D_M as the diagonal matrix where $D_{ii} = 1$ for $i = 1, 2$ and $D_{33} = \det(UV^\top)$. Then we define a mapping $O : M \mapsto R_i$ such that it performs the *orthogonalization* of the matrix M:

$$R_i = U_M D_M V_M^\top \tag{20}$$

The orthogonalisation of a matrix M is an operation that projects it to the nearest element in $\text{SO}(3)$ in the least-squares sense (Levinson et al. (2020)).

The cross-product has the following general property. If R is an orthogonal matrix in $\mathbb{R}^{3\times 3}$, then we have the identity:

$$(RV_1) \times (RV_2) = R(V_1 \times V_2) \tag{21}$$

Note that since V_{R_i} transforms according to $\rho_{E_r}^{\text{out}} = \varphi_r \oplus \varphi_r$, a block-diagonal matrix where the blocks are two copies of the three-dimensional representation φ_r, V_1 and V_2, and according to the identity in equation 21 also V_3 transform according to φ_r (φ_r is orthogonal). Since V_1, V_2 and V_3 are the columns of M, all the columns of the matrix M transform according to φ_r.

To prove that equation 15 is satisfied, we need to prove that the orthogonalization mapping $O : M \mapsto R_i$ is equivariant, that is

$$O(\varphi_r M) = \varphi_r O(M). \tag{22}$$

We can prove this by using the previously mentioned property of orthogonalization of a matrix that it is an operation that projects a matrix to the nearest element in $\text{SO}(3)$ in the least-squares sense:

$$O(M) = \underset{R \in \text{SO}(3)}{\arg\min} ||R - M||_F^2 \tag{23}$$

We assume that the singular values of M and are distinct, so that we can assume that the singular value decomposition is unique. Then we can prove equation 22 via a proof by contradiction as follows:

Suppose that $O(M) = \arg\min_{R\in\text{SO}(3)} ||R-M||_F^2 = R$ and suppose that equation 22 does not hold, that is $O(\varphi_r M) = R' \neq \varphi_r R$. Recall that we assumed that the singular value decomposition was unique, thus we can assume that R' and R are the unique elements of SO(3) minimizing the norm in equation 22. Using that the Frobenius norm is invariant under multiplication with orthogonal matrices, we can derive

$$||R' - \varphi M||_F^2 < ||\varphi R - \varphi M||_F^2 \tag{24}$$

$$||\varphi_r^\top (R' - \varphi M)||_F^2 < ||\varphi_r^\top (\varphi R - \varphi M)||_F^2 \tag{25}$$

$$||\varphi^\top R' - \varphi^\top \varphi M||_F^2 < ||\varphi^\top \varphi R - \varphi^\top \varphi M||_F^2 \tag{26}$$

$$||\varphi^\top R' - M||_F^2 < ||R - M||_F^2. \tag{27}$$

This is a contradiction, since we assumed that $R = \arg\min_{R\in\text{SO}(3)} ||R - M||_F^2$. So we must have that equation 22 holds.

[2]The cross product $V_3 = V_1 \times V_2$ is given by $[v_2v_6 - v_3v_5 \ \ v_3v_4 - v_1v_6 \ \ v_1v_5 - v_2v_4]^\top$. The direction of V_3 is perpendicular to the plane formed by V_1 and V_2, according to the right-hand rule

F METRICS

Pose prediction error The reconstructed volume and therefore also the rotations predicted by the encoder given images in the dataset differ from their ground truth (used for the simulation of the data) by some global rotation $R^{\mathrm{G}} \in \mathrm{SO}(3)$. Therefore, before determining the pose prediction error we first have to find an estimation for R^{G} and use it to re-align the predicted poses to the ground truth poses. We follow a method based on publicly available code provided by the authors of Levy et al. (2022a). In this method, the mean squared error (MSE) and median squared error (MedSE) for a given dataset $\mathcal{D} = \{I_1, \ldots, I_N\}$ of N images are determined as follows. For every $I_i \in \mathcal{D}$, we write $R_i^{\text{pred}} := E_r(I_i)$ for the rotation predicted by the encoder, and R_i^{gt} for the rotation that was used in the simulation of the dataset to generate I_i. Let REL be the set containing the relative rotations that align each of the predicted rotations to their corresponding ground-truth rotation:

$$\text{REL} := \{R_i^{\text{rel}} \mid \text{for all } 1 \leq i \leq N \text{ such that } R_i^{\text{rel}} R_i^{\text{pred}} = R_i^{\text{gt}}\}. \tag{28}$$

We then choose the rotations in REL that minimize the mean squared Frobenius norm and the median squared Frobenius norm between the aligned predicted rotations and the ground truth rotations to calculate the MSE and MedSE:

$$\text{MSE} := \min_{R \in \text{REL}} \frac{1}{N} \sum_{i}^{N} \| R R_i^{\text{pred}} - R_i^{\text{gt}} \|_{\text{F}}^2, \text{ and} \tag{29}$$

$$\text{MedSE} := \min_{R \in \text{REL}} \text{median} \left(\| R R_1^{\text{pred}} - R_1^{\text{gt}} \|_{\text{F}}^2, \ldots, \| R R_N^{\text{pred}} - R_N^{\text{gt}} \|_{\text{F}}^2 \right). \tag{30}$$

Resolution The Fourier Shell Correlation (FSC) is a widely utilized method in cryo-EM to assess the resolution of 3D reconstructions. It quantifies the degree of correlation between two independent reconstructions obtained from different sets of data.

Let V_1 and V_2 represent two 3D reconstructed volumes in real space. The Fourier transforms of these volumes are denoted as $\hat{V}_1$ and $\hat{V}_2$. The FSC is computed as a function of the spatial frequency, and for a given shell in the Fourier space (defined by a specific frequency magnitude $|k|$), it is given by:

$$\text{FSC}(|k|) = \frac{\sum_{|k|} \hat{V}_1(k) \cdot \overline{\hat{V}_2(k)}}{\sqrt{\sum_{|k|} |\hat{V}_1(k)|^2 \cdot \sum_{|k|} |\hat{V}_2(k)|^2}} \tag{31}$$

where $\overline{\hat{V}_2(k)}$ denotes the complex conjugate of $\hat{V}_2(k)$.

The FSC curve, plotted as a function of spatial frequency, provides insight into the consistency of structural information between the two reconstructions at different resolutions. A higher FSC value indicates a higher degree of correlation, suggesting better agreement between the reconstructions. The resolution at which the FSC curve drops to a threshold value (commonly set to 0.143 or 0.5) is typically taken as an indicator of the resolution of the reconstructed volume. In this work, we determine the resolution by comparing the reconstructed volume with the corresponding ground truth volume used to create the simulated data. We set the threshold to 0.5 to determine the resolution of the volume.

G EXPLORATION OF CONVERGENCE PROCESS OF STANDARD PIPELINE

As described in Section 5.1, the standard cryoAI pipeline optimized with a symmetric loss function shows a convergence pattern in which there is an initial decline in both MedSE and MSE, with MedSE approaching zero fast and MSE plateauing, then sharply decreasing (Figure 2).

Given how we define the MSE and the MedSE (see Section F, Equation 29 and Equation 30), this suggests that a subset of at least half of the predicted poses can be aligned to the ground truth with a low error relatively quickly, while the the other subset of poses show a higher error with this alignment.

Internally consistent subsets Consider a subset of images $S \subseteq \mathcal{D}$ of the full dataset $\mathcal{D}$. For any image I_i in S and its corresponding predicted rotation R_i^{pred}, we define the relative rotation R_i^{rel} such that $R_i^{\text{rel}} R_i^{\text{pred}} = R_i^{\text{gt}}$. This rotation aligns the predicted rotations with the ground truth rotation. If the encoder of the model is consistent on the subset S, we expect R_i^{rel} to align the predicted rotations of all images in S with their ground truths ($R_i^{\text{rel}} R_j^{\text{pred}} = R_j^{\text{gt}}$ for all j where I_j is in S). Therefore, if an image I_i is part of an internally consistent subset S_i, counting the number of images I_j where $R_i^{\text{rel}} R_j^{\text{pred}} \approx R_j^{\text{gt}}$ gives an approximate size of S_i. In mathematical terms, for a given i, we can define S_i as follows:

$$S_i = \{I_j : ||R_i^{rel} R_j^{pred} - R_j^{gt}|| < \epsilon\} \subseteq D. \tag{32}$$

If the encoder makes prediction for all images in a dataset $\mathcal{D}$ with low error, each subset S_i would be equal to the entire dataset ($|S_i| = |\mathcal{D}|$ for every I_i in $\mathcal{D}$). Conversely, if the encoder is in a state where it classifies half of the images into one internally consistent subset S and the other half into another subset S', then $|S| = |S'| = \frac{1}{2}|\mathcal{D}|$. We found that in states where the MSE was high whereas the MedSE had already decreased, the latter was the case.

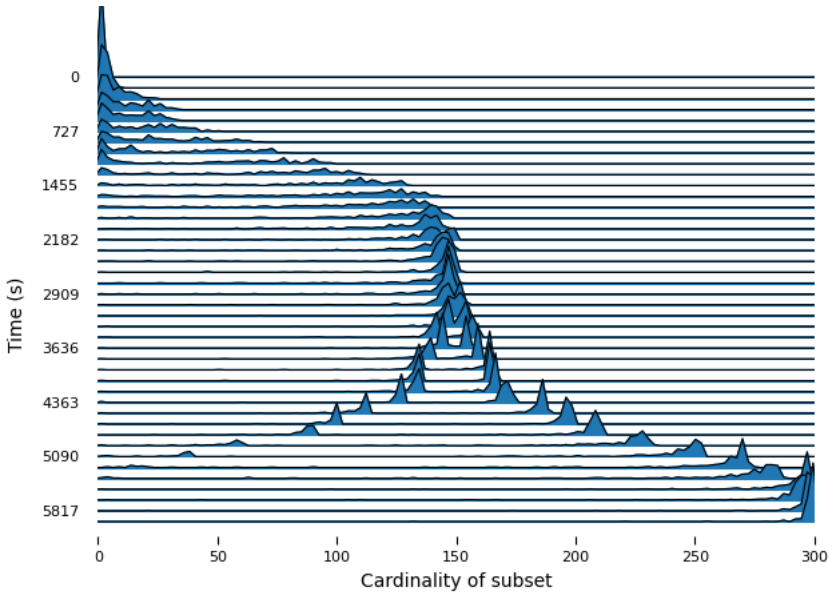


Figure 7: Joy plot with on each horizontal line the distribution of the cardinalities of subsets $|S_i|$ for the first 300 images in our dataset $\mathcal{D}$ at a given time step.

We show this in Figure 7, where we illustrate the distribution of $|S_i|$ for the first 300 images in the training dataset $\mathcal{D}$, observed over time in a run exhibiting the characteristic trajectory mentioned earlier. We notice that after some time, every image belongs to a group of size 150, which is half of the data points. This indicates the existence of two internally consistent subsets of images. This state persists for a while, after which one subset starts shrinking while the other grows until it encompasses all data points. The moment one subset begins to shrink and the other to grow coincides with the decrease in MSE.

Projective plane visualization To explore the convergence process of the encoder, we developed a way to visualize the way in which the encoder predicts poses on the projective plane.

Let $\mathcal{D} = \{I_1, I_2, \ldots, I_n\}$ represent the set of images from a cryo-EM experiment. We define the equivalence relation $\sim$ on $\mathcal{D}$ such that $I_i \sim I_j$ if and only if there exists some $g \in \mathrm{O}(2)$ where $I_j = L_g[I_i]$ (where L_g is the left action of g, see footnote 1). This relation leads to the formation of the set of equivalence classes $\mathcal{D}/\sim$.

Since each image I_i in $\mathcal{D}$ corresponds to an element R_i in SO(3), each element in $\mathcal{D}/\sim$ can be seen as part of the real projective plane $\mathrm{P}\mathbb{R}^2 \cong$ SO(3)/O(2). The projective plane can be imagined as the unit sphere S^2 where each pair of opposite points is identified. Recall that each image has

a corresponding pose $R_i = (x_i, y_i, z_i)$ in SO(3), with z_i being the axis of projection in the image formation process, also known as the 'viewing direction'. Each equivalence class $[I_i]$ in $\mathcal{D}/\sim$ then corresponds to a point on the projective plane, where the viewing direction z_i is one of such paired points.

We make the projective plane visualizations as shown in Figure 8, by projecting poses onto the xy-plane as shown *after first aligning them with the ground truth poses* in the same way as we would to to determine the MSE (see F). We do this for both the ground truth poses and the poses predicted by the encoder. To visualize how the predicted poses differ from their ground truth pose, we color each of the predicted poses with the color that their corresponding ground truth pose has according to a colormap on the xy-plane.

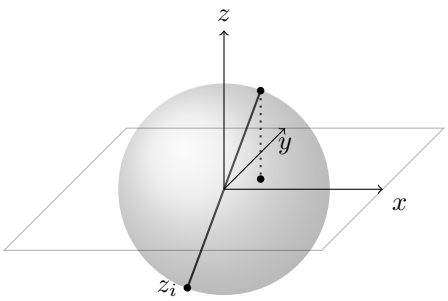
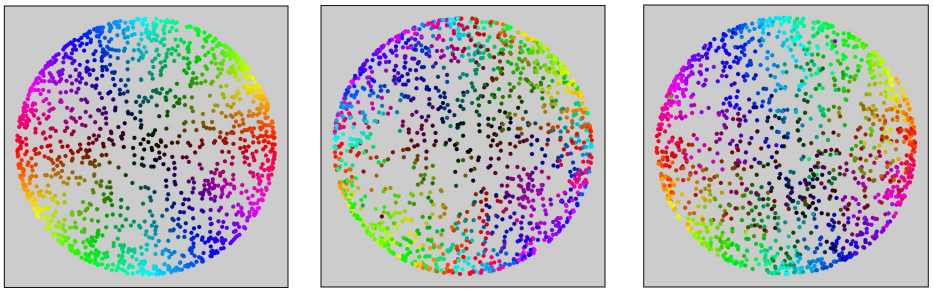


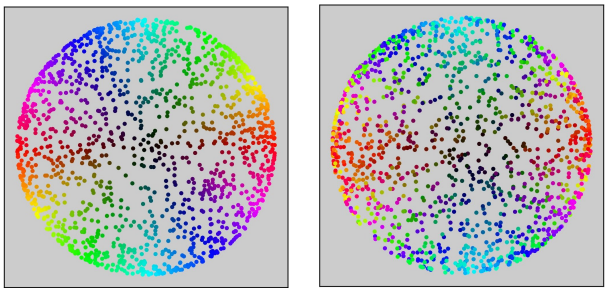
Figure 8: We interpret a pose $R_i = (x_i, y_i, z_i)$ as a point on the projective plane and visualize this as depicted. If $z_i = (z_{i1}, z_{i2}, z_{i3})$, then we project R_i to the point $(\text{sgn}(z_{i3})z_{i1}, \text{sgn}(z_{i3})z_{i1})$ on the plane. Here sgn is the function that returns the sign of its input.

In Figure 10 and Figure 9 we show examples of patterns in the projective plane visualizations that occur for the different datasets in states where the MSE is high, and the MedSE is low. For both structures, a pattern is observable where approximately half of the point are mirrored with respect to some line in the xy-plane. From these patterns in the projective plane alone, interpreting how the encoder misclassifies subsets of poses is challenging. However, the patterns indicate that the internally consistent subsets of data points are sometimes distinguishable within the projective plane.



Ground truth poses | Pattern 1 of predicted poses | Pattern 2 of predicted poses

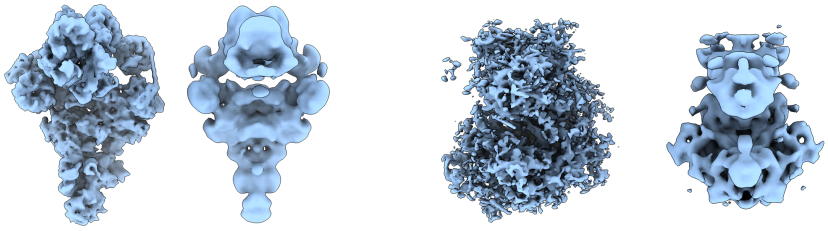
Figure 9: Patterns observed in projective plane visualisations for runs trained on datasets generated with the 80S ribosome structure.



Ground truth poses | Pattern of predicted poses

Figure 10: Patterns observed in projective plane visualisations for runs trained on datasets generated with the spliceosome structure.

Spurious planar symmetries In addition, the discussed states corresponded to the decoder reconstructing volumes that have a spurious planar symmetry. In Figure 11 we show reconstructions with such symmetries together with to their corresponding ground truth volumes. This phenomenon that is also discussed in Levy et al. (2022a), in which it was shown that a symmetric loss function prevents the model from getting stuck in states showing these symmetries. Our exploration shows that even with a symmetric loss function, these states occur.



Spliceosome structure 80S ribosome structure

Figure 11: Visualization of ground truth volume and reconstructed volume showing spurious planar symmetry for the spliceosome structure (left) and the 80S ribosome structure (right). For both structures, the ground truth volume is shown on the left, and the reconstructed volume on the right. Visualizations were made using ChimeraX Meng et al. (2023)

Upon comparing the reconstructions showing planar symmetry with the projective plane visualizations as shown in Figure 9 and Figure 10, we find that for the patterns in which a subset of points is mirrored with respect to a line in the xy-plane, this line corresponds to the plane of symmetry of the reconstructed volume.

We hypothesize that the patterns in the projective plane and the planar symmetry in the reconstructed volume can be explained by the fact that for volumes that have a planar symmetry, some viewing directions yield very similar projections, even though they would yield different projections in a non-symmetric volume. We show an example of this in Figure 12. This enables the pipeline to reconstruct images in the dataset with a low loss in two disctinct ways. The pipeline in stuck in a state reconstructing a volume with planar symmetry when the encoder is equally likely to predict each of these two. This explains the two equally sized internally consistent subsets of poses.

Interpretation of results The cryoAI authors (Levy et al. (2022a)) have noted that without a symmetric loss function, the pipeline could remain in local minima featuring reconstruction showing spurious planar symmetries for extended periods. We find that even with a symmetric loss function, the standard model can temporarily get stuck in similar local minima. In these states, due to planar symmetry, two distinct poses can generate the same projection, unlike in non-symmetric volumes. The exploratory study also reveals that when the model is in a state reconstructing a symmetric volume, the poses predicted by the encoder form two equally sized groups, each internally consistent but inconsistent with each other. We hypothesize that these groups represent the two distinct ways of generating the same projection in a volume showing spurious planar symmetry.

We hypothesize that the symmetric loss helps the model escape states in which it is reconstructing volumes with spurious planar symmetries as follows. Utilizing a symmetric loss function involves augmenting each batch with a rotated (or in the case of a mirror loss, mirrored) duplicate of each image in the batch. Both versions of the image pass through the model, but the model is only supervised on the best reconstruction. Recall that we observed that, when reconstructing a volume showing a spurious planar symmetry, two distinct poses can generate the same projection. We expect that at least sometimes, rotating (or mirroring) an image causes the encoder to switch from predicting one pose to the other that can generate the same projection. If, by chance, one aspect of the symmetry in the reconstructed volume provides marginally better reconstructions for a specific image, the model may begin to prefer this aspect, and as such 'escape' the symmetric state.

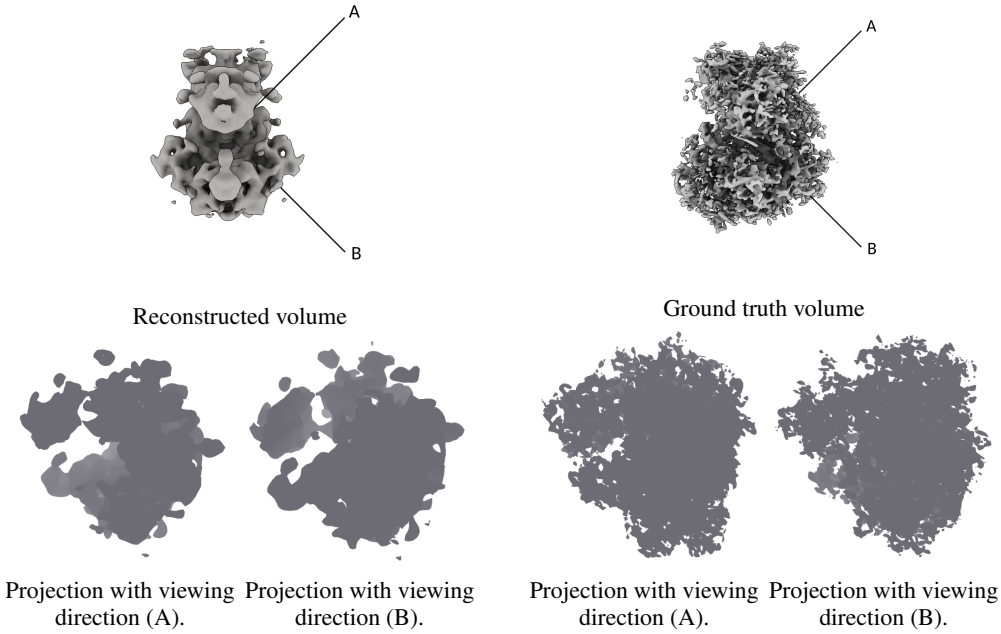


Figure 12: A reconstructed volume showing planar symmetry with two viewing directions (A) and (B) is shown on the top left. Viewing direction (B) is the viewing direction that is obtained if (A) is mirrored along the plane of symmetry and multiplied by -1. On the top right, the ground truth volume is shown with the same viewing directions. For the volume showing planar symmetry, viewing directions (A) and (B) generate projections that look almost identical due to planar symmetry. On the other hand, the ground truth volume does not show planar symmetry and the projections generated by both viewing directions look different. Visualizations were made with ChimeraX Meng et al. (2023).