electronic reprint

STRUCTURAL BIOLOGY

ISSN: 2059-7983
journals.iucr.org/d

End-to-end orientation estimation from 2D cryo-EM images

Ruyi Lian, Bingyao Huang, Liguo Wang, Qun Liu, Yuewei Lin and Haibin Ling

Acta Cryst. (2022). D**78**, 174–186

IUCr Journals
CRYSTALLOGRAPHY JOURNALS ONLINE

Author(s) of this article may load this reprint on their own web site or institutional repository provided that this cover page is retained. Republication of this article or its storage in electronic databases other than as specified above is not permitted without prior permission in writing from the IUCr.

For further information see https://journals.iucr.org/services/authorrights.html

STRUCTURAL BIOLOGY

ISSN 2059-7983

End-to-end orientation estimation from 2D cryo-EM images

Ruyi Lian,[a]* Bingyao Huang,[a] Liguo Wang,[b] Qun Liu,[c] Yuewei Lin[d] and Haibin Ling[a]*

[a]Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA, [b]Laboratory for Biomolecular Structure, Brookhaven National Laboratory, Upton, NY 11973, USA, [c]Biology Department, NSLS-II, Brookhaven National Laboratory, Upton, NY 11973, USA, and [d]Computational Science Initiative, Brookhaven National Laboratory, Upton, NY 11973, USA. *Correspondence e-mail: rulian@cs.stonybrook.edu, hling@cs.stonybrook.edu

Received 14 June 2021
Accepted 5 November 2021

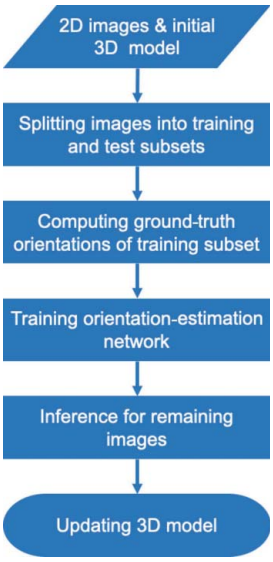
Edited by K. R. Vinothkumar, National Centre for Biological Sciences-TIFR, India

Keywords: 3D reconstruction; image processing; single-particle cryo-EM.

Cryo-electron microscopy (cryo-EM) is a Nobel Prize-winning technique for determining high-resolution 3D structures of biological macromolecules. A 3D structure is reconstructed from hundreds of thousands of noisy 2D projection images. However, existing 3D reconstruction methods are still time-consuming, and one of the major computational bottlenecks is recovering the unknown orientation of the particle in each 2D image. The dominant methods typically exploit an expensive global search on each image to estimate the missing orientations. Here, a novel end-to-end supervised learning method is introduced to directly recover the missing orientations from 2D cryo-EM images. A neural network is used to approximate the mapping from images to orientations. A robust loss function is proposed for optimizing the parameters of the network, which can handle both asymmetric and symmetric 3D structures. Experiments on synthetic data sets with various symmetry types confirm that the neural network is capable of recovering orientations from 2D cryo-EM images, and the results on a real cryo-EM data set further demonstrate its potential under more challenging imaging conditions.

1. Introduction

Cryo-electron microscopy (cryo-EM) is a powerful technique for determining the structures of biological macromolecules at atomic or near-atomic resolution. In single-particle cryo-EM, a central problem is to reconstruct the 3D structure of a macromolecule from 10^4–10^7 noisy 2D projection images extracted from multiple micrographs. The orientation of the particle captured in each 2D image is unknown because the 3D particle adopts a random orientation in the ice layer. The orientation-estimation step is critical in the 3D reconstruction process because the 3D structure can be reconstructed based on the Fourier slice theorem (Bracewell, 1956) once the unknown orientations of the 2D images have been recovered. Gupta *et al.* (2021) showed that it is possible to skip the orientation-estimation step and directly reconstruct the 3D structure in a generative adversarial network (GAN) framework. Despite these efforts, orientation estimation remains one of the most crucial steps in mainstream reconstruction solutions.



Orientation estimation is typically very difficult and time-consuming. A popular conventional method is to determine relative orientations based on common lines in Fourier space (Vainshtein & Goncharov, 1986; Van Heel, 1987). However, detecting common lines from extremely noisy images is in

electronic reprint

itself very challenging (Wang *et al.*, 2013; Greenberg & Shkolnisky, 2017; Bendory *et al.*, 2020). Another popular conventional approach adopts an iterative framework which alternatively refines the 3D structure and orientation estimations (Scheres, 2012*a*). The orientation of each image is estimated independently by comparing the image with all possible projections of the 3D model. Despite its robustness and accuracy, this process is computationally expensive.

Recently, deep-learning-based methods have shown promising results in solving many cryo-EM-related problems, including particle picking (Wang *et al.*, 2016; Zhu *et al.*, 2017; Bepler, Morin *et al.*, 2019; Wagner *et al.*, 2019; Al-Azzawi *et al.*, 2019; McSweeney *et al.*, 2020), denoising (Bepler *et al.*, 2020; Palovcak *et al.*, 2020; Li *et al.*, 2021; Huang *et al.*, 2020), 3D reconstruction (Gupta *et al.*, 2020, 2021; Zhong *et al.*, 2019) *etc*. The neural networks can learn to extract useful features from cryo-EM images to facilitate cryo-EM tasks. For orientation estimation, methods based on a variational autoencoder (VAE) have been proposed to encode in-plane orientations in a latent space (Bepler, Zhong *et al.*, 2019; Miolane *et al.*, 2020; Bibas *et al.*, 2021). However, it is nontrivial to extend these VAE-based methods to recover 3D orientations (Zhong *et al.*, 2021). To fully recover 3D orientations, Xie *et al.* (2020) utilized a *k*-nearest-neighbor network to refine the orientations obtained from a global projection-matching method. Banjac *et al.* (2021) proposed a two-step deep-learning-based method in which the network estimates distances between pairs of cryo-EM images, instead of the 3D orientations for each image. Jiménez-Moreno *et al.* (2021) further proposed dividing the 3D orientations into non-overlapping subsets and training separate neural networks to classify whether the unknown orientation belongs to the corresponding subset.

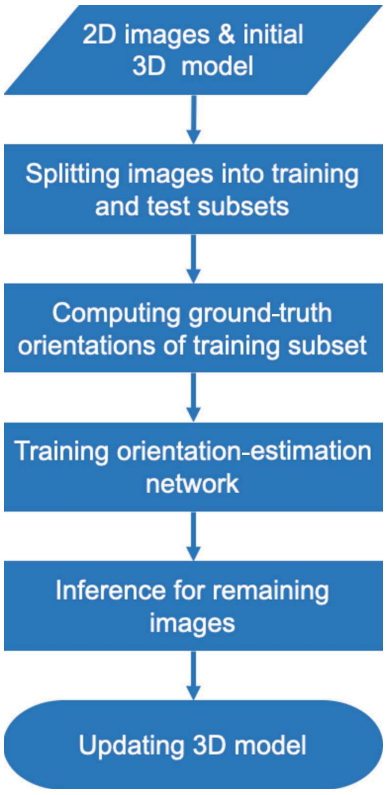


Figure 1
Overview of our proposed supervised learning method for orientation estimation when an initial 3D model is available. We used a subset of the cryo-EM images as training images and obtained the ground-truth orientations by aligning the images with the given initial model. After training, the network can directly infer the unknown orientations for the remaining images. The 3D model can then be updated based on the estimations.

In this work, we present a new supervised learning method to recover the unknown orientations directly from 2D cryo-EM images. Following previous work (Jiménez-Moreno *et al.*, 2021), we assume that an initial model is available and that all of the observed cryo-EM images are 2D projections of the same 3D model. Unlike the supervised learning method (Banjac *et al.*, 2021), our method is end-to-end trainable. We utilize a single neural network to estimate 3D orientations for given input images. For symmetric particles, our network will output one of the orientations that are equivalent with respect to the symmetry, since the symmetry type can be imposed later in 3D reconstruction.

We evaluate our network on synthetic data sets with various symmetry types, and the results confirm that our network is capable of recovering the orientations from synthetic cryo-EM images. Evaluation on a real data set further demonstrates that our method works promisingly under more challenging imaging conditions.

2. Methods

2.1. Supervised learning pipeline

To estimate the orientations of the particle from each 2D cryo-EM image in 3D refinement, a conventional method typically utilizes a global search to find the orientation that best aligns the observed image with the 3D model. To accelerate the orientation-estimation process, we propose a supervised learning method, which is illustrated in Fig. 1. Instead of a brute-force search, we utilize a neural network to directly determine estimated orientations from input images.

To train the network in a supervised way, a subset of the images are used as a training set, and each training image is annotated with one ground-truth orientation obtained by the conventional method. For a symmetric particle, one cryo-EM image may correspond to multiple orientations, and the design of our training objective allows the network to learn efficiently from the single annotation. After training, the neural network is utilized to recover the orientations from the remaining particle images.

2.2. Image-formation model for cryo-EM images

Our method utilizes the image-formation model for cryo-EM images. In cryo-EM research, a 3D model V is often represented as the mapping from a 3D coordinate to the electron density at this 3D point,

$$V : \mathbb{R}^3 \rightarrow \mathbb{R}. \tag{1}$$

electronic reprint

A 2D cryo-EM particle image X (in real space) extracted from a micrograph can be modeled as a noisy projection of V along the imaging axis (*i.e.* the z axis; Bendory *et al.*, 2020),

$$X_{R,t}(p_x, p_y) = g * \int_{\mathbb{R}} V(R^T\mathbf{p} + t)\,\mathrm{d}r_z + \text{noise}, \tag{2}$$

where g is the point-spread function of the microscope, $\mathbf{p} = (p_x, p_y, p_z)^T$ is a 3D point, $R \in SO(3)$ is the orientation of V and $t = (t_x, t_y, 0)$ is an in-plane translation corresponding to imperfect centering of V during the particle-picking process.

Correspondingly, in Fourier space the generative process for image $\hat{X}$ from $\hat{V}$ can be modeled as

$$\hat{X}_{R,t}(k_x, k_y) = \hat{g}S(t)A(R)\hat{V}(k_x, k_y) + \varepsilon, \tag{3}$$

where $\hat{g}$ is the Fourier transform of the point-spread function g, called the contrast-transfer function (CTF) of the microscope, $S(t)$ is a phase-shift operator corresponding to an in-plane translation t in real space, $A(R)\hat{V} = \hat{V}[R^T(\cdot, \cdot, 0)^T]$ is a linear slice operator corresponding to the combination of rotation R and linear projection along the z axis in real space, and ε is frequency-dependent noise in Fourier space.

The Fourier slice theorem (Bracewell, 1956) states that the two-dimensional Fourier transform of a projection image is the restriction of the three-dimensional Fourier transform of the 3D particle to a planar central slice perpendicular to the viewing direction,

$$\mathcal{F}_2 PR \circ V = SR \circ \mathcal{F}_3 V, \tag{4}$$

where $\mathcal{F}_2$ and $\mathcal{F}_3$ denote the Fourier transform over $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively, $R \circ V = V(R^T\mathbf{p})$, P is the tomographic projection operator along the z axis and S denotes the restriction operator to the xy plane.

Based on the Fourier slice theorem, one can reconstruct the 3D structure V by estimating its 3D Fourier transform $\hat{V}$ using 2D projections with corresponding orientations and in-plane translations. The idea is popularly used in reconstruction from cryo-EM particles (Scheres, 2012*a*; Punjani *et al.*, 2017).

In this work, we focus on orientation estimation, *i.e.* given an observed 2D particle image X and a 3D model V in real space, we aim to recover the 3D orientation $R \in SO(3)$ in (2) that best aligns X with V with ground-truth in-plane translation. For symmetric particles, the orientation is not unique. Thus, we define the set $\mathcal{S}(X)$ of orientations R that all correspond to the image X as

$$\mathcal{S}(X) = \{R \in SO(3) | X_{R,t} = X\}, \tag{5}$$

where t is the ground-truth in-plane translation. We propose the use of a neural network to approximate the following mapping $f(\cdot, \cdot)$

$$f : X, V \rightarrow r(R), \tag{6}$$

where $r(R) \in \mathbb{R}^d$ is a d-dimensional vector as a represention of $R \in \mathcal{S}(X)$. If V is symmetric, our network will output one of the equivalent orientations with respect to the symmetry.

2.3. Representation of orientations

Before designing the architecture of the orientation-estimation network, we need to determine how to efficiently represent 3D orientations $R \in SO(3)$ as network output.

Compared with a 3×3 rotation matrix, the axis–angle representation is a more compact representation which is used in cryo-EM software packages including *cryoSPARC* (Punjani *et al.*, 2017). Suppose that the axis of rotation is a unit vector $\mathbf{v}$ and the magnitude of the rotation about the axis is the angle θ; the rotation can then be represented as a 3D rotation vector $\mathbf{r}$,

$$\mathbf{r} = \theta\mathbf{v}, \tag{7}$$

and the corresponding 3×3 rotation matrix R can be computed by Rodrigues' rotation formula (Murray *et al.*, 2017) as

$$R = I_3 + \sin\theta[\mathbf{r}]_\times + (1 - \cos\theta)[\mathbf{v}]_\times^2, \tag{8}$$

where I_3 is a 3×3 identity matrix and $[\mathbf{v}]_\times$ is a skew-symmetric operator of vector $\mathbf{v}$.

Another popular compact representation is a quaternion, which is a four-dimensional unit vector. Given an axis–angle representation $\mathbf{r} = \theta\mathbf{v}$, the corresponding quaternion $\mathbf{q}$ can be computed as

$$\mathbf{q} = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{v}\right)^T, \tag{9}$$

and it is easy to check that $\|\mathbf{q}\|_2 = 1$. The corresponding 3×3 rotation matrix R can be computed as

$$R = \begin{bmatrix} 1 - 2(q_3^2 - q_4^2) & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & 1 - 2(q_2^2 + q_4^2) & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & 1 - 2(q_2^2 + q_3^2) \end{bmatrix} \tag{10}$$

for $\mathbf{q} = (q_1, q_2, q_3, q_4)^T$.

In this work, we adopt the quaternion representation and let our network regress a four-dimensional vector for an input image. To fulfill the constraint that $\|\mathbf{q}\|_2 = 1$, we simply normalize the network output. Our proposed network can also be modified to regress other representations such as axis–angle representations, and we compare the network performance with different representations in our experiments.

2.4. Network architecture

For a given 3D model V, we approximate the mapping f in (6) via an orientation-estimation network $\hat{f}_\Theta(\cdot, \cdot)$, where Θ represents the learnable network parameters.

Fig. 2 illustrates the architecture of our network. Given a 2D cryo-EM image X in real space as the input, the network outputs an unnormalized four-dimensional vector, which will be normalized as the estimated quaternion $\mathbf{q}_{\text{pred}}$. The size of the input is fixed as 128×128.

To efficiently extract low-level image features including edges, corners, color conjunctions *etc.* and reduce the training time, we utilize the first three layers from the VGG16 network (Simonyan & Zisserman, 2014) as our feature extractor, which is pretrained on a large public natural image repository named

electronic reprint

ImageNet (Deng *et al.*, 2009). The parameters are fine-tuned during training. Since the input cryo-EM image has only one channel, we normalize the values to (−1, 1) and obtain a three-channel image by repeating the channel. The VGG16 network layers then extract 256 feature maps with spatial dimensions 16 × 16.

Two additional convolutional layers are used to process the extracted low-level features and generate 2048 feature maps with spatial dimensions 4 × 4. These feature maps are downsampled to 2 × 2 by a max pooling layer. Finally, an orientation-regression module composed of several fully connected layers regresses an unnormalized four-dimensional vector from the convolutional features. We explicitly normalize the vector to a unit vactor.

2.5. Loss functions

To train the orientation-estimation network $\hat{f}_\Theta$, we need to define proper loss functions to optimize the learnable parameters Θ by penalizing the distance between the network prediction and the ground truth. A straightforward loss function is the L_1/L_2 distance between the predicted normalized quaternion and the ground truth. However, this loss function does not utilize the underlying geometric structure of $SO(3)$.

According to the geodesic distance between two quaternions $\mathbf{q}_1$, $\mathbf{q}_2$,

$$d_g(\mathbf{q}_1, \mathbf{q}_2) = 2\cos^{-1}(|\langle \mathbf{q}_1, \mathbf{q}_2 \rangle|), \tag{11}$$

we can define a computation-efficient loss function, called quaternion-distance (QD) loss, as

$$\ell_{\mathrm{QD}}(\mathbf{q}_{\mathrm{pred}}, \mathbf{q}_{\mathrm{gt}}) = 1 - \langle \mathbf{q}_{\mathrm{pred}}, \mathbf{q}_{\mathrm{gt}} \rangle^2, \tag{12}$$

where $\mathbf{q}_{\mathrm{pred}}$ is the network prediction and $\mathbf{q}_{\mathrm{gt}}$ is the ground truth.

During training, the symmetries of 3D particles can cause ambiguity because images with similar appearances may correspond to different annotated orientations. Naively regressing the annotated orientations often ends up with predicting an orientation that is closest to all results in the symmetry group (Manhardt *et al.*, 2019). To handle this issue, some previous methods restricted the range of orientations used for training (Kehl *et al.*, 2017; Rad & Lepetit, 2017) or explicitly incorporated the symmetry type into the loss functions (Park *et al.*, 2019; Labbé *et al.*, 2020). Instead, we propose another loss function called reprojection loss to ensure that the 2D projection along the network prediction is consistent with the observed image. Since we focus on orientation estimation, we remove the CTF and noise corruption from the projection process (2) and compute the simplified result as

$$\tilde{X}_{R,t} = \int_{\mathbb{R}} V(R^T\mathbf{p} + t)\,\mathrm{d}r_z. \tag{13}$$

The reprojection loss is then defined as the L_1 distance between the simplified projections,

$$\ell_{\mathrm{reproj}} = \|\tilde{X}_{R_{\mathrm{pred}},t} - \tilde{X}_{R_{\mathrm{gt}},t}\|_1, \tag{14}$$

where R_{pred} and R_{gt} are the 3 × 3 rotation matrices corresponding to q_{pred} and q_{gt}, respectively, and t can be any valid in-plane translation as long as the particle is projected within the 2D image. When the 3D model V is asymmetric, both reprojection loss and QD loss will achieve their optimal values at the same unique orientation. When V is symmetric, unlike QD loss, reprojection loss will not penalize estimations that are equivalent with respect to the symmetry of V.

Although reprojection loss can automatically handle the particle symmetry, the forward computation and back-propagation of reprojection loss is much more complex than that of QD loss. Using only reprojection loss is prone to a local minimum. To make the training process more efficient, we further propose the use of a weighted sum of QD loss and reprojection loss as our training objective,

$$\ell = w\ell_{\mathrm{QD}} + (1 - w)\ell_{\mathrm{reproj}}, \tag{15}$$

where $w \in (0, 1)$ is a weighting factor to balance these two losses. Adding QD loss can accelerate the convergence of training, especially when the 3D particle V is asymmetric. Intuitively, w should be close to 1 when the 3D model does not

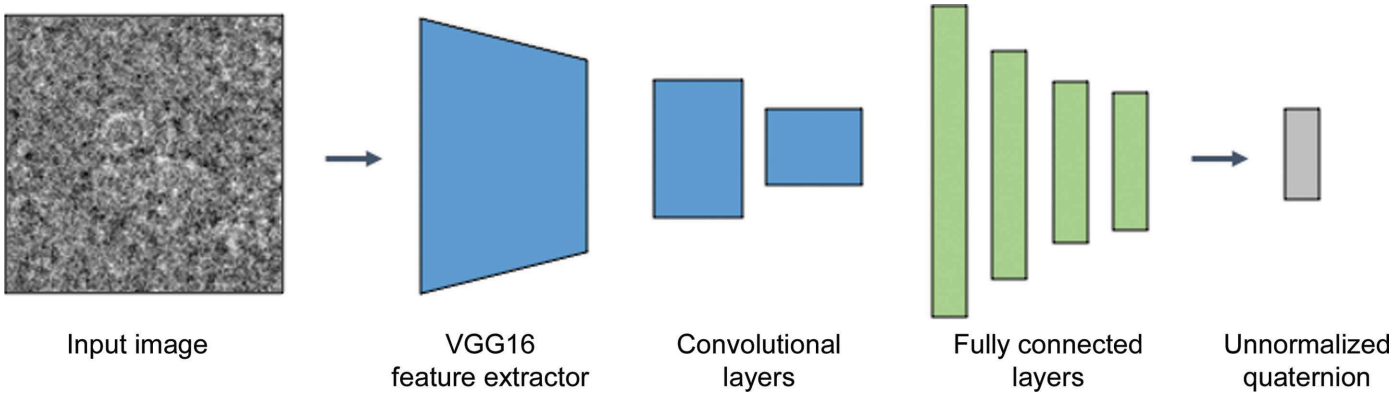


Figure 2
The architecture of our proposed orientation-estimation network. We first normalized the input image in the range (−1, 1) and converted it to a three-channel image by repeating the channel. The features extracted by convolutional layers are passed to a 2 × 2 max pooling layer and then input to the fully connected layers to regress a four-dimensional vector, which will be normalized to be a unit quaternion as the estimated orientation.

electronic reprint

have symmetry and close to 0 when the 3D model has any type of symmetry, which is also confirmed in our experiments.

2.6. Data sets

To test the feasibility of our supervised learning method for orientation estimation, we first generated two noiseless synthetic data sets from PDB models following the cryo-EM image-formation model (2) named EMPIAR-10061-simu and EMPIAR-10028-simu. To generate EMPIAR-10061-simu, we utilized a known β-galactosidase structure. The 3D cryo-EM density map of EMPIAR-10061-simu was obtained by fitting a 3 Å resolution map from PDB entry 5a1a (Bartesaghi *et al.*, 2015) in *UCSF Chimera* (Pettersen *et al.*, 2004) and low-pass filtering it to a 12 Å resolution map in *cryoSPARC* (Punjani *et al.*, 2017). We then uniformly sampled 3D rotation matrices but set the translation as zero. We also randomly sampled the defocus parameters of the CTF. In this way, we obtained 10 000 noiseless synthetic projection images. For EMPIAR-10028-simu, we utilized PDB entries 3j7a and 3j79, which are the small subunit and the large subunit of the *Plasmodium falciparum* 80S ribosome, respectively (Wong *et al.*, 2014). We fitted one single 4 Å resolution density map in *UCSF Chimera* from the PDB models and again low-pass filtered it to a 12 Å resolution map in *cryoSPARC*. We then followed the same sampling procedure to generate 10 000 noiseless synthetic projection images.

Besides using PDB models, we also generated projections from an initial model produced by *cryoSPARC* to create another synthetic data set called EMPIAR-10025-simu. Specifically, we picked 8111 good particle images from the subset of 20 movies in the EMPIAR-10025 *Thermoplasma acidophilum* 20S proteasome data set (Campbell *et al.*, 2015) in *cryoSPARC*, and ran *ab initio* reconstruction in *cryoSPARC* to obtain a 12 Å resolution initial model. We then uniformly sampled 3D rotation matrices in $SO(3)$ and 2D translations in $(t_{\min}, t_{\max}) \times (t_{\min}, t_{\max})$, where $t_{\min}$ and $t_{\max}$ are the minima and maxima of the translations estimated from the real cryo-EM images, respectively. For the CTF parameters, we directly used the estimations from the real cryo-EM images. Finally, we generated 8000 projection images from the *ab initio* model with uniformly sampled orientations and translations.

To see the performance of our method on sets of real cryo-EM images, we generated another data set called EMPIAR-10025-real. We reused the same 3D model and the first 8000 real cryo-EM images from the *ab initio* reconstruction process which was used to generate EMPIAR-10025-simu. The signal-to-noise ratio was 0.18. For simplicity, we treated the 3D orientations and 2D translations estimated in the *ab initio* reconstruction process as the ground-truth pose, which was close enough to the real ground truth for a proof of concept.

Since the size of our network is 128×128, for all data sets we resized the box size of the 3D models to 128^3, so the synthetic projections are naturally 128×128. We also resized the real cryo-EM images to 128×128. We split all of the image sets into training sets and test sets using a 4:1 ratio.

An important property of these data sets is that they have different levels of symmetry. For EMPIAR-10028-simu, the 3D model is asymmetric. For EMPIAR-10061-simu, the 3D model has $D2$ symmetry. For EMPIAR-10025-simu and EMPIAR-10025-real, the 3D model has $D7$ symmetry. We believe that varying symmetry levels are necessary to test the robustness of orientation-estimation methods.

2.7. Implementation details

We implemented our method in *PyTorch* (Paszke *et al.*, 2017). For all experiments, we trained the network on a single Nvidia TITAN Xp GPU for 2000 iterations with a batch size of 24, and it took about 10 min to finish. We used the Adam optimizer (Kingma & Ba, 2014) and fixed the ℓ_2 penalty factor to 10^{-4}. We also decayed the learning rate by a factor of 5 after the first 1600 iterations. Without specification, the initial learning rate η_0 was set as a default value, *i.e.* 5×10^{-5}. We also implemented a differentiable projection process in *PyTorch*, which was used to generate projections for synthetic data sets and compute the reprojection loss during training. For all experiments, we set t in (13) as the ground-truth in-plane translation when computing reprojection loss.

2.8. Evaluation metrics

For all data sets, we evaluated the network performance on test images by computing the root-mean-square error (RMSE) between the simplified projections $\tilde{X}_{R_{\text{pred}},t}$ and $\tilde{X}_{R_{\text{gt}},t}$ as

$$\text{RMSE} = \left(\frac{\sum_{i=1}^{N} \| \tilde{X}^{(i)}_{R_{\text{pred}},t} - \tilde{X}^{(i)}_{R_{\text{gt}},t} \|_2^2}{N} \right)^{1/2}, \tag{16}$$

where N is the number of test images and $i = 1, 2, \ldots, N$ is the index of the test image. This metric, called the reprojection RMSE, is well defined for data sets with any types of symmetry.

3. Results

3.1. Orientation estimation from synthetic cryo-EM images

We first evaluated our network on three synthetic data sets to see whether the network could recover the orientations from synthetic projections, and how the performance is affected by the weighting factor w in our training objective. Thus, on each synthetic data set, we trained the network with various w and evaluated the network predictions for test images in terms of reprojection RMSE. The quantitative results are presented in Fig. 3 and Table 1.

On EMPIAR-10028-simu, the asymmetric synthetic data set for the *P. falciparum* 80S ribosome, the quantitative results show that the network can generate reasonable predictions when $w \geq 0.6$. The network achieves the lowest reprojection RMSE (0.1097) when $w = 0.8$, and the visualization in Fig. 4 confirms that the network predictions are close to the ground-truth orientations.

On the two symmetric synthetic data sets, the quantitative results show that the network is capable of recovering the

electronic reprint

orientations when $w \leq 0.4$. The lowest reprojection RMSEs on EMPIAR-10061-simu (β-galactosidase) and EMPIAR-10025-simu (the *T. acidophilum* 20S proteasome) are 0.1469 and 0.1025, respectively, which are both achieved when $w = 0.1$. The visualizations of the best performances in Fig. 4 also confirm that the projections along network predictions are highly consistent with the input images, even though the data sets have certain types of symmetry.

From the results on synthetic data sets, we can see that the network trained with weighted loss can handle different types of symmetry. Specifically, the QD weighting factor w should have a large value for asymmetric particles and a small value for symmetric particles.

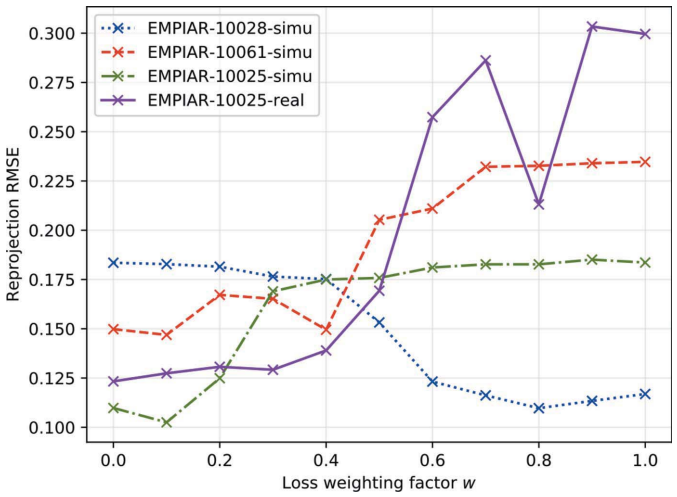


Figure 3
The network performance in terms of reprojection RMSE on test sets, with different weighting factors w, on four data sets.

Table 1
The best performance of our network on each data set in terms of reprojection RMSE.

Data set	w	RMSE
EMPIAR-10028-simu	0.8	0.1097
EMPIAR-10061-simu	0.1	0.1469
EMPIAR-10025-simu	0.1	0.1025
EMPIAR-10025-real	0.0	0.1233

3.2. Orientation estimation from real cryo-EM images

Next, we evaluated our network on a real data set, EMPIAR-10025-real (the *T. acidophilum* 20S proteasome), to see whether our network could recover orientations under real-life imaging conditions. We again trained the network with various w and the quantitative results are shown in Fig. 3 and Table 1. The network can output acceptable predictions when $w \leq 0.4$, which is consistent with our observations on symmetric synthetic data sets. The network achieves the lowest reprojection RMSE (0.1233) when $w = 0.0$, and the visualization in Fig. 4 shows that the network is capable of recovering the orientations from real cryo-EM images.

3.3. Cross-validation tests for w

To determine the optimal values of w in our weighted training objective, we ran fivefold cross-validation tests on our four data sets. For each data set, we split the images into five equal-sized subsets. For w from 0.0 to 1.0 with a fixed step size of 0.1, in the ith ($1 \leq i \leq 5$) test the ith subset was then used as validation data while the remaining subsets were used to train our network. Finally. we report the average reprojection RMSEs on the validation data in Fig. 5.

For EMPIAR-10028-simu (the *P. falciparum* 80S ribosome), the asymmetric data set, the cross-validation results showed that the optimal w is 0.8, and our network can

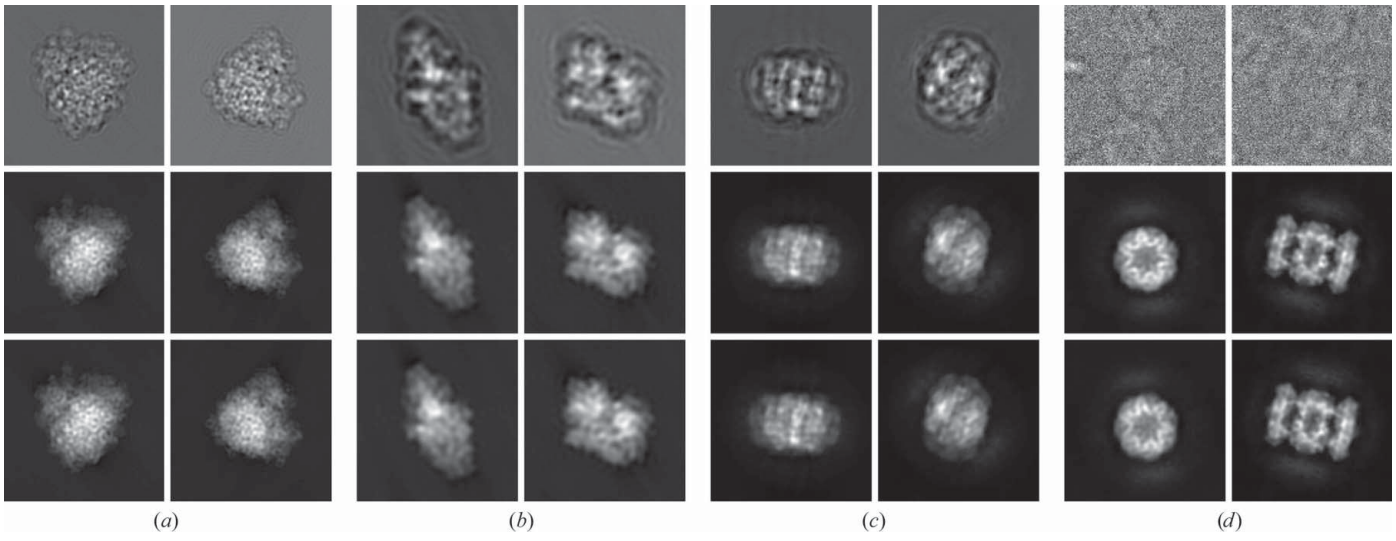


Figure 4
Visualization of the network performance on the test set of four data sets. (*a*) EMPIAR-10028-simu with $w = 0.8$. (*b*) EMPIAR-10061-simu with $w = 0.1$. (*c*) EMPIAR-10025-simu with $w = 0.1$. (*d*) EMPIAR-10025-real with $w = 0.0$. For each data set, the first row shows the network inputs, the second row shows the corresponding simplified projections along the ground-truth orientations and the third row shows the simplified projections along the network predictions.

electronic reprint

Table 2
The performance of our network with different levels of noise in terms of reprojection RMSE.

Data set	w	Noiseless	SNR = 1.0	SNR = 0.7	SNR = 0.4	SNR = 0.1
EMPIAR-10028-simu	0.8	0.1097	0.1300	0.1315	0.1426	0.1846
EMPIAR-10061-simu	0.1	0.1469	0.1559	0.1826	0.1548	0.2309
EMPIAR-10025-simu	0.0	0.0982	0.1130	0.1124	0.1125	0.1165

generate reasonable predictions when $w \leq 0.6$. For EMPIAR-10061 (β-galactosidase) with $D2$ symmetry the optimal w was 0.1, while for EMPIAR-10025-simu and EMPIAR-10025-real (the *T. acidophilum* 20S proteasome) with $D7$ symmetry the optimal w was 0.0. Besides, our network is capable of recovering the orientations when $w \leq 0.4$ for the three symmetric data sets.

We further visualized the evolution of training losses and evaluation metrics on each data set with the corresponding optimal w in Figs. 6 and 7, which demonstrate that the training process can converge after 2000 iterations. For EMPIAR-10028-simu, the asymmetric data set, QD loss decreases rapidly especially during the first 1000 iterations; thus, setting a large weight for QD loss can lead to rapid convergence. For the three symmetric data sets the QD loss oscillates severely, while the reprojection loss steadily decreases.

3.4. Influence of different noise levels

To see how the level of noise affects the performance of our method, following the image-formation model (2) we perturbed the images of the three synthetic data sets with different levels of additive Gaussian noise. Specifically, we trained our network with four different signal-to-noise ratios (SNRs): 1.0, 0.7, 0.4 and 0.1. The quantitative results are shown in Table 2. For EMPIAR-10028-simu (the *P. falciparum* 80S ribosome), the quantitative results in Table 2 and visualization in Fig. 8 show that the network can generate reasonable predictions when the SNR varies from 1.0 to 0.4. For EMPIAR-10061-simu (β-galactosidase), the quantitative results in Table 2 and visualization in Fig. 9 show that the network can generate reasonable predictions when SNR = 1.0 and SNR = 0.4. For EMPIAR-10025-simu (the *T. acidophilum* 20S proteasome), the quantitative results in Table 2 and visualization in Fig. 10 show that the network can generate reasonable predictions when the SNR varies from 1.0 to 0.1. From the results with different noise levels, we can see that our network is capable of learning orientation estimation from noisy cryo-EM images.

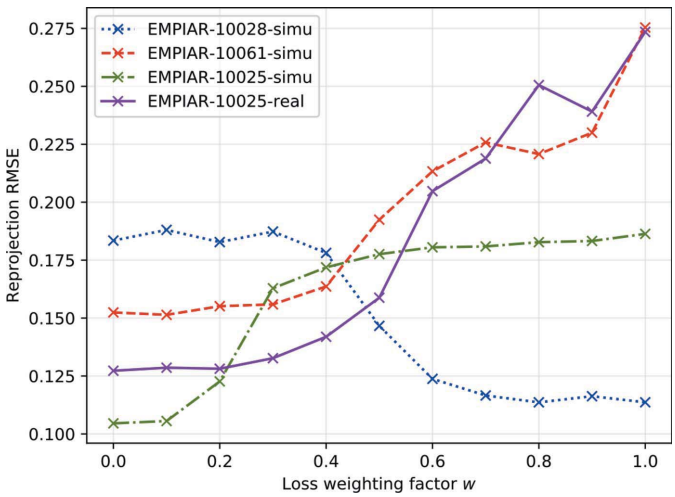


Figure 5
The network performance in terms of reprojection RMSE for fivefold cross-validation tests on four data sets.

Table 3
Comparison of the performance of the orientation-estimation network trained with different loss-weighting factors w, orientation representations and initial learning rates η_0 on the EMPIAR-10028-simu data set.

w	Representation	η_0	RMSE
0.8	Axis–angle	1×10^{-4}	0.1800
		5×10^{-5}	0.1364
		1×10^{-5}	0.1698
	Quaternion	1×10^{-4}	0.1136
		5×10^{-5}	0.1097
		1×10^{-5}	0.1728
1.0	Axis–angle	1×10^{-4}	0.2090
		5×10^{-5}	0.1406
		1×10^{-5}	0.1717
	Quaternion	1×10^{-4}	0.1154
		5×10^{-5}	0.1169
		1×10^{-5}	0.1750

3.5. Comparison of different representations of 3D orientations

To see how the representations of 3D orientations affect our supervised learning method, we compare the network performance on the asymmetric EMPIAR-10028-simu data set (the *P. falciparum* 80S ribosome) with axis–angle representation and quaternion representation using different initial learning rates and loss-weighting factors. For axis–angle representation, we reduced the dimension of the network output layer from four to three, and restricted each component in $(-\pi, \pi)$. The quantitative results are shown in Table 3. We can see that the best performance with quaternion representation is much better than the best performance with axis–angle representation in terms of reprojection RMSE. Besides, the performance with quaternion representation is more robust under different initial learning rates η_0 and loss-weighting factors w.

3.6. 3D reconstruction using network estimation

Here, we show the results of 3D reconstruction using orientations estimated by our network. We obtained the 3D structure from 1600 test images of the EMPIAR-10025-real data set (the *T. acidophilum* 20S proteasome) using ground-truth orientations (Fig. 11*a*). We also ran the reconstruction process using estimations from our network trained with $w = 0.0$, 0.4 and 0.8, respectively (Figs. 11*b*–11*d*). Since we are focusing on the orientation-estimation step at this stage of development, all of the reconstructions were performed by a

electronic reprint

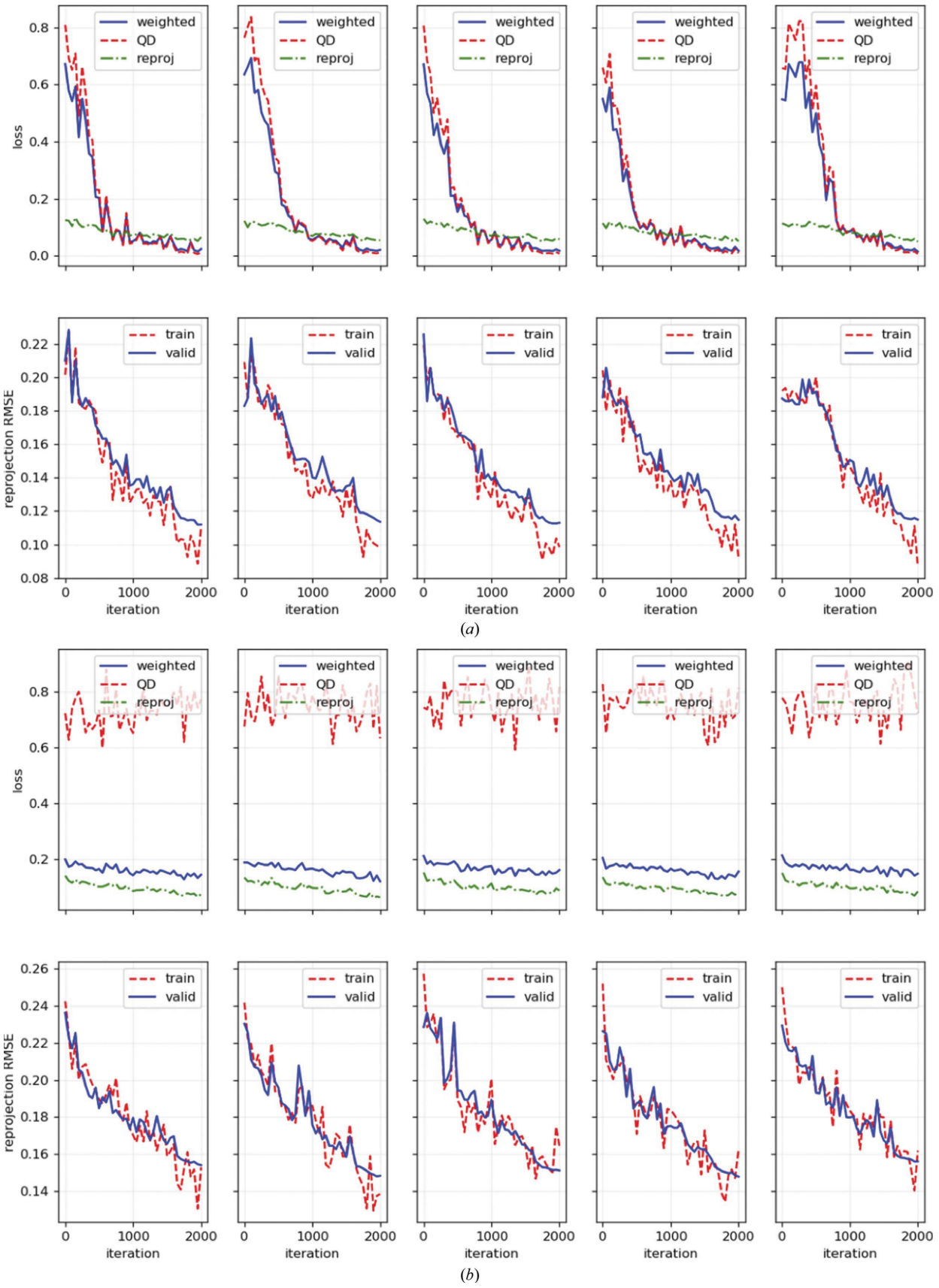


Figure 6
The curves of training losses and evaluation metrics in fivefold cross-validation. (*a*) EMPIAR-10028-simu ($w = 0.8$). (*b*) EMPIAR-10061-simu ($w = 0.1$). The first row shows the curves of our weighted training objective, as well as QD loss and reprojection loss. The second row shows the reprojection RMSE of the batch of training images and the evaluation data.

electronic reprint

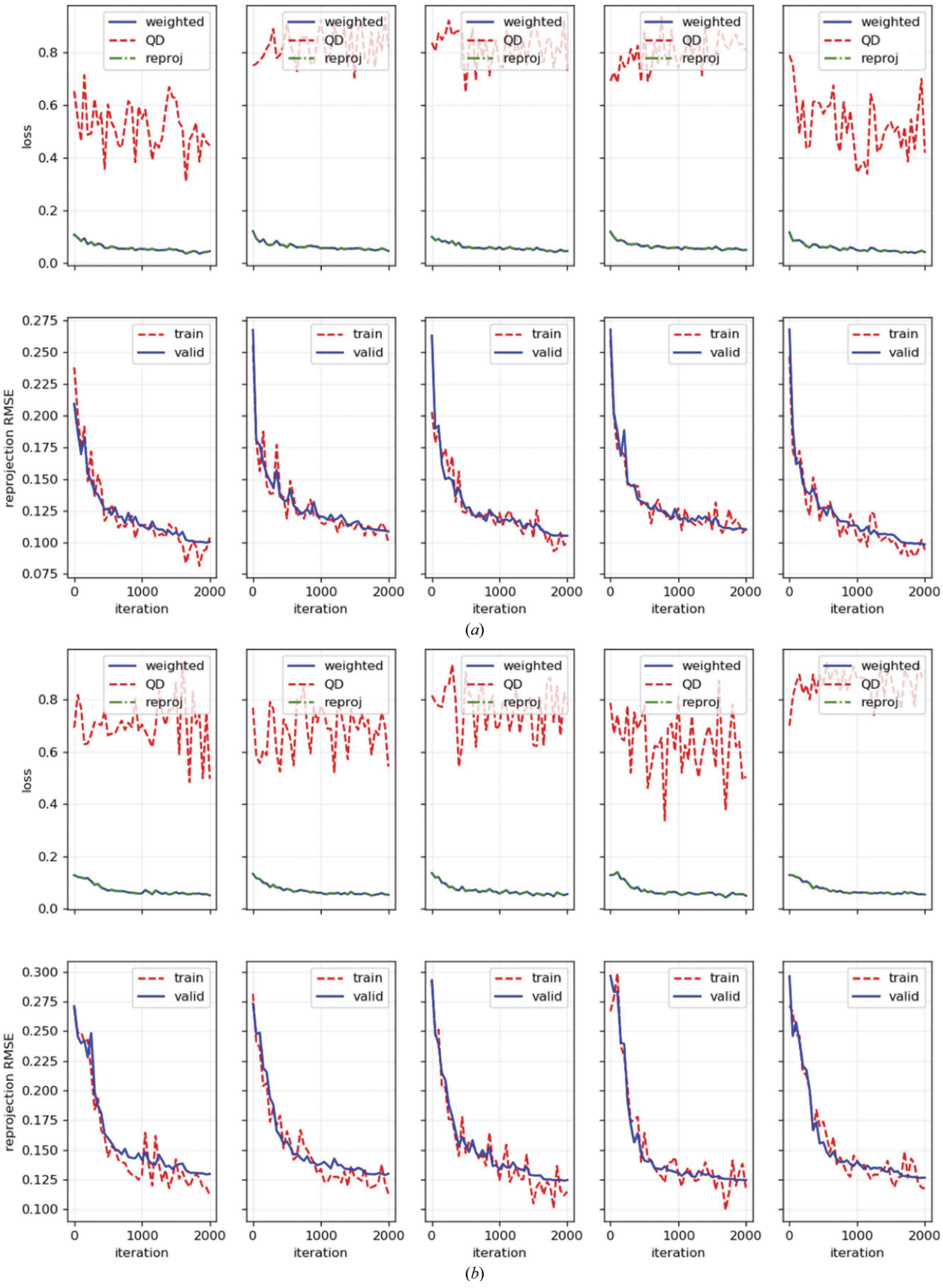


Figure 7
The curves of training losses and evaluation metrics in fivefold cross-validation. (*a*) EMPIAR-10025-simu ($w = 0.0$). (*b*) EMPIAR-10025-real ($w = 0.0$). The first row shows the curves of our weighted training objective, as well as QD loss and reprojection loss. The second row shows the reprojection RMSE of the batch of training images and the evaluation data.

electronic reprint

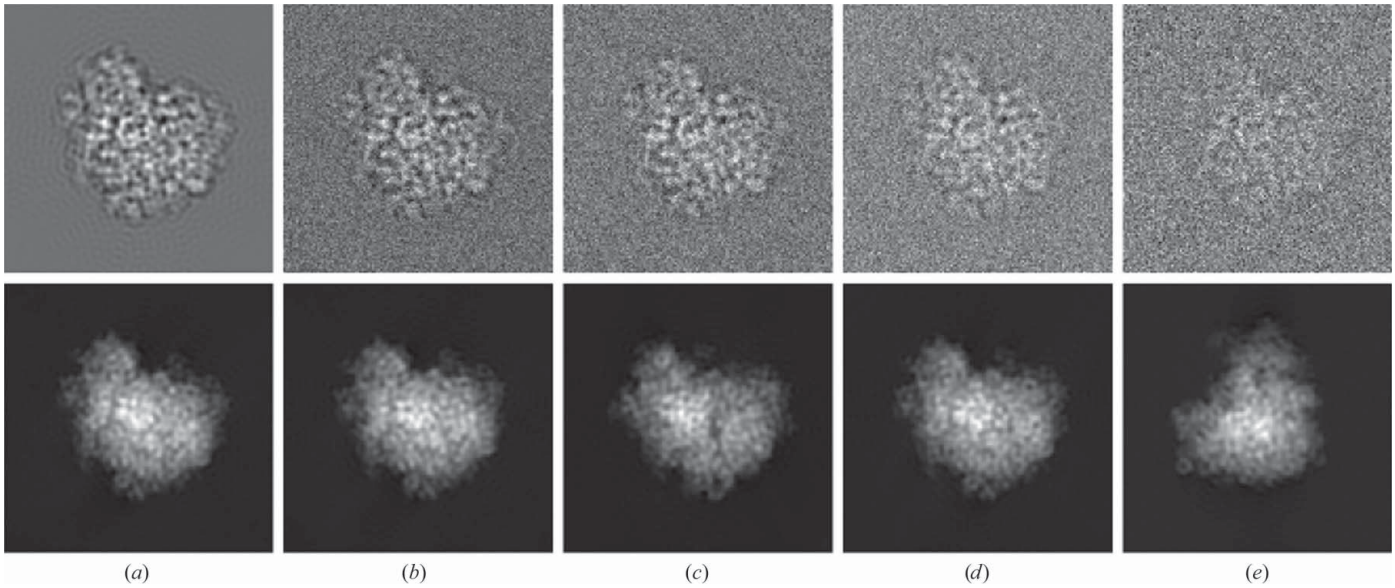


Figure 8
Visualization of the network performance on the EMPIAR-10028-simu data set with different noise levels. The first row shows the network inputs and the second row shows the simplified projections along the network predictions. (*a*) Noiseless input. (*b*) SNR = 1.0. (*c*) SNR = 0.7. (*d*) SNR = 0.4. (*e*) SNR = 0.1. For all experiments, we trained our network using $w = 0.8$.

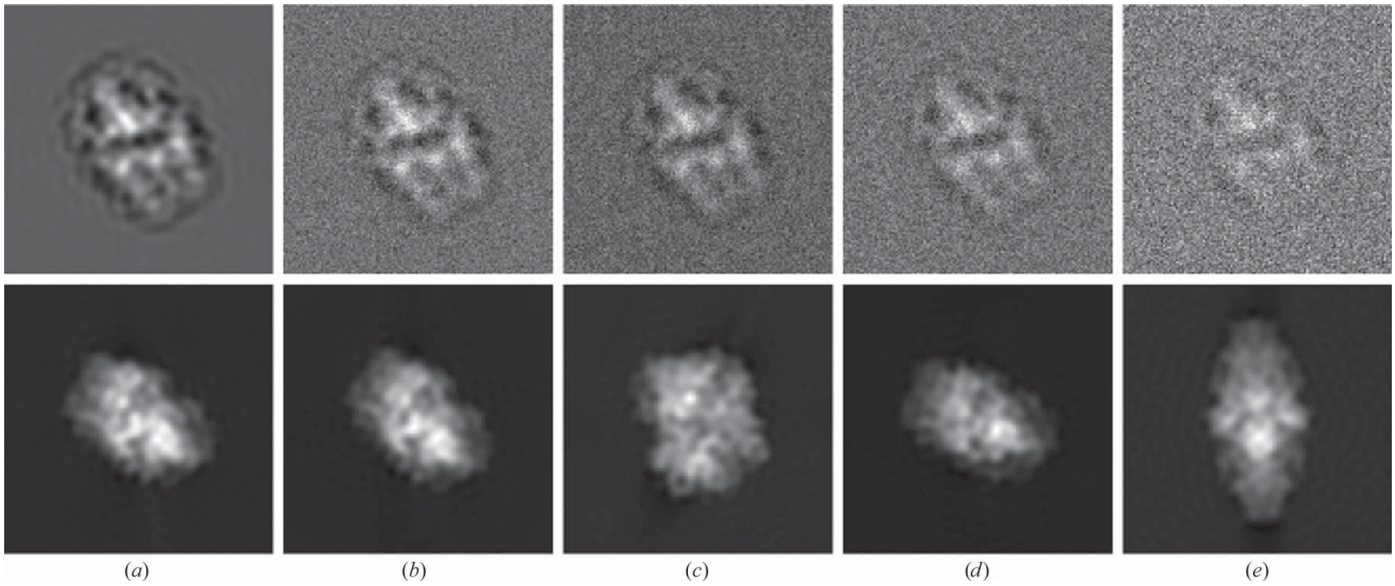


Figure 9
Visualization of the network performance on the EMPIAR-10061-simu data set with different noise levels. The first row shows the network inputs and the second row shows the simplified projections along the network predictions. (*a*) Noiseless input. (*b*) SNR = 1.0. (*c*) SNR = 0.7. (*d*) SNR = 0.4. (*e*) SNR = 0.1. For all experiments, we trained our network using $w = 0.1$.

simple back-projection algorithm in *RELION* (Scheres, 2012*b*) instead of more robust iterative methods. Besides, we back-project each 2D image along the orientations that are equivalent to the given one with respect to *D*7 symmetry.

Fig. 11(*e*) shows the Fourier shell correlation (FSC) curves between the reconstructions using network estimations and the reconstruction using ground-truth orientations. The estimation from our network trained with $w = 0.0$, which has the lowest reprojection RMSE, results in the closest reconstruction (Fig. 11*b*) to the ground truth (Fig. 11*a*). When we use the larger value of $w = 0.4$, the orientation estimation becomes worse in terms of reprojection RMSE and the quality of the reconstruction (Fig. 11*c*) also decreases. When we further increase the value of w to 0.8, the accuracy of the orientation estimation is even lower and the quality of the reconstruction (Fig. 11*d*) significantly declines.

4. Discussion

4.1. Orientation estimation

In this work, we have explored the possibility of using a single neural network to estimate orientations from 2D

electronic reprint

cryo-EM images. We conducted experiments on four data sets with various symmetry types and imaging conditions, and the results demonstrate that after being trained for 2000 iterations on $\leq$8000 images, our network can achieve a decent performance in terms of reprojection RMSE.

Our orientation-estimation method can be exploited during the 3D refinement process, where an initial 3D model is available. To prepare the training data, we can run global searching on a small subset of the cryo-EM images to obtain the ground-truth orientations. After training, the network can directly output the orientation estimations for the remaining images. Since the 3D refinement process typically updates the orientation estimation and 3D reconstruction iteratively, we claim that our network only needs to be trained from scratch

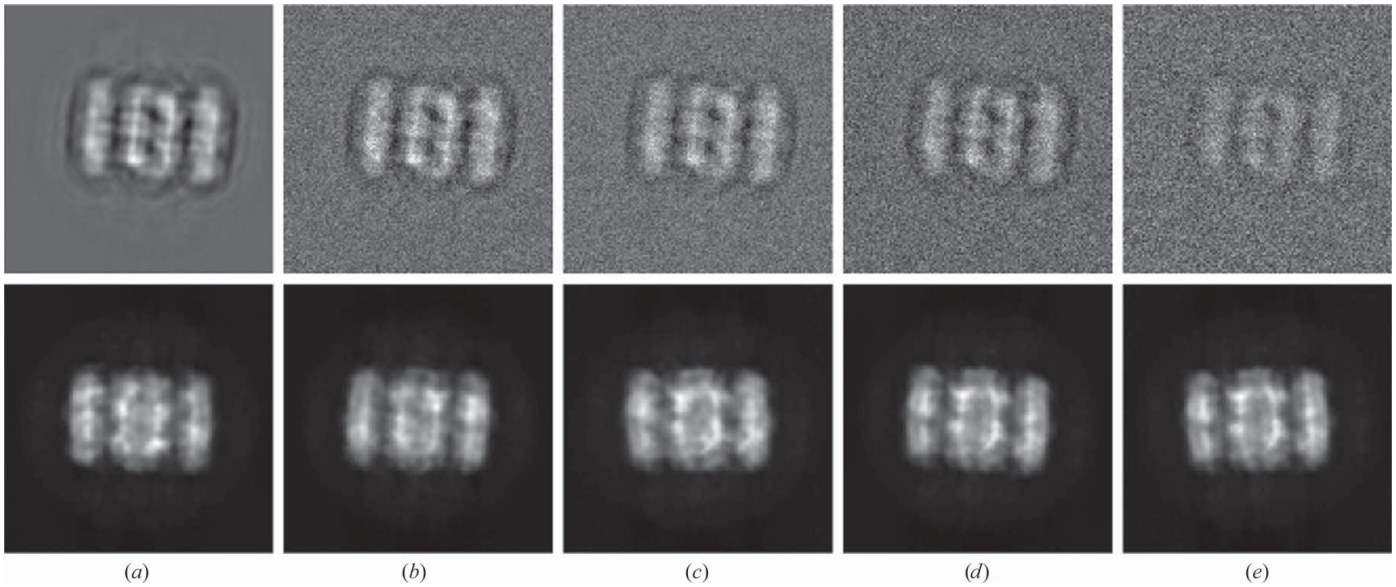


Figure 10
Visualization of the network performance on the EMPIAR-10025-simu data set with different noise levels. The first row shows the network inputs and the second row shows the simplified projections along the network predictions. (*a*) Noiseless input. (*b*) SNR = 1.0. (*c*) SNR = 0.7. (*d*) SNR = 0.4. (*e*) SNR = 0.1. For all experiments, we trained our network using w = 0.0.

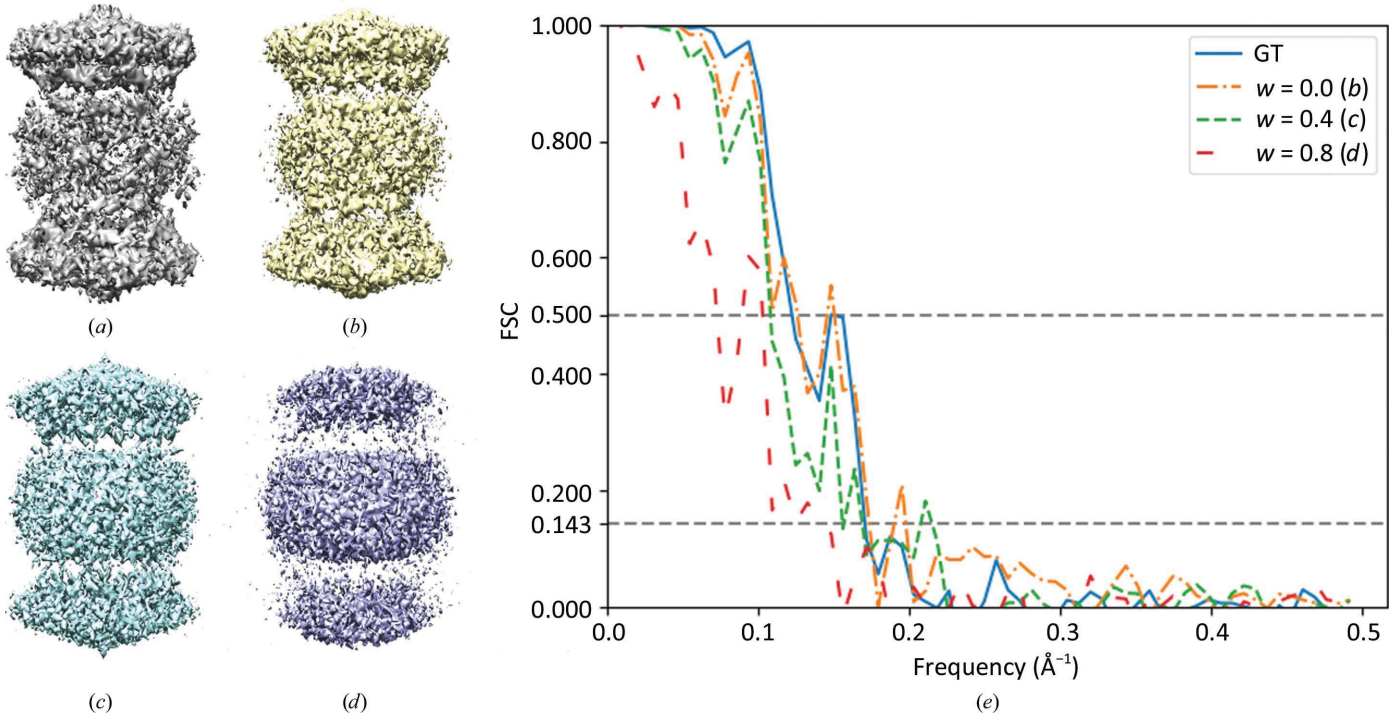


Figure 11
3D reconstruction results from test images with *D*7 symmetry on the EMPIAR-10025-real data set. (*a*) Reconstruction using ground-truth orientations. (*b*, *c*, *d*) Reconstructions using estimations from out network trained with w = 0.0, 0.4 and 0.8, respectively. (*e*) The FSC curve of ground-truth reconstruction (*a*), as well as the FSC curves between the reconstructions using network estimations and the ground truth.

electronic reprint

in the first refinement iteration. For the subsequent refinement iterations, our network can be quickly fine-tuned. This will significantly reduce the total training time and accelerate the 3D refinement process.

4.2. Training objective

Our training objective is a weighted combination of two loss terms. The first term, called QD loss, directly regresses the orientations based on geodesic distance. The second term, called reprojection loss, encourages the projections along the estimated orientations to be consistent with the input images. The computation of reprojection loss (14) is much complex than that of QD loss (12). However, for symmetric particles the reprojection loss allows the network to output one of the orientations that are equivalent to the ground truth with respect to the symmetry. In this way, our training objective can implicitly handle the symmetry of the given 3D model without restricting the range of orientations used for training.

The weighting factor w in the training objective is used to balance the QD loss and the reprojection loss. Our experiments show that the network is robust within a large range of weighting factors w (Fig. 3). For asymmetric data sets we can use a large value such as $w \geq 0.6$, and for data sets with potential symmetry we can set a small value such as $w \leq 0.4$. Although using only reprojection loss is prone to a local minimum due to the complexity of the loss function, for EMPIAR-10025-real, the real data set with $D7$ symmetry, the experimental results show that simply setting $w = 0.0$ is ideal to achieve optimal results. In practice, to find a suitable w for training a new data set one can test values from 0.0 to 1.0 with a fixed step size of 0.1.

When computing the reprojection loss, we treat the in-plane translations as known values and use the ground-truth values in our experiments. One may extend our method to recover in-plane translations simultaneously. This is because the output layer of our network can be modified to estimate in-plane translations as well, and the reprojection loss can be directly used to jointly optimize the prediction for 3D orientation and in-plane translation.

4.3. Evaluation metric and reconstruction quality

For evaluation, we compute the reprojection RMSE for the orientations estimated from our network. Similar to the reprojection loss used in our training objective, this evaluation metric measures the inconsistency between the projections along the network predictions and the input images. Intuitively, lower inconsistency will lead to higher quality 3D reconstructions. The reconstruction results on the EMPIAR-10025-real data set (Fig. 11) also indicate that orientation estimations with lower reprojection RMSE result in better 3D reconstructions. Thus, it is reasonable to improve the network performance in terms of reprojection RMSE in order to achieve better 3D reconstructions after recovering the orientations.

Since the ultimate goal of recovering orientations is to obtain a high-resolution 3D density map, a future research direction is to incorporate the metric for reconstruction quality into our end-to-end trainable pipeline. One may modify the training objective by adding another loss term based on FSC.

4.4. Handling real cryo-EM images

In practice, the signal-to-noise ratio (SNR) of real cryo-EM images is typically far below 1 (Frank & Al-Ali, 1975). The extremely low SNR poses a challenge for orientation estimation because it is difficult to distinguish particles from noise.

In the experiment on the EMPIAR-10025-real data set, we input the real cryo-EM images into our network without denoising. The result (Figs. 3 and 4) indicates that our network is promising for the recovery of orientations from noisy cryo-EM images. One can also add a denoising module to our orientation-estimation framework, so that our network can focus on the signal of the particle, and the training process will be more efficient. This can be performed by conventional methods such as low-pass filtering or recent deep-learning-based methods (Bepler *et al.*, 2020; Palovcak *et al.*, 2020; Li *et al.*, 2021; Huang *et al.*, 2020).

In our current implementation, the image size of the network input is fixed at 128×128. This may not be sufficient for processing the images of large complexes such as viruses, or running iterations of 3D refinement that require high-resolution images. A future research direction is to efficiently process larger network inputs without significantly increasing the computational cost.

5. Concluding remarks

Recovering 3D orientations from hundreds of thousands of 2D cryo-EM images is still a time-consuming step in the 3D reconstruction pipeline. To efficiently estimate the unknown orientations, we have proposed a novel end-to-end trainable framework with a robust weighted loss function. We have also tested the method on synthetic and real images. Our method may be extended to also recover the in-plane translations and incorporate them into the iterative 3D reconstruction pipeline.

Funding information

This work was supported in part by the SBU–BNL Seed Grant Program and National Science Foundation (NSF) Grant 1814745. QL was supported by the US Department of Energy, Office of Science, Office of Biological and Environmental Research as part of the Quantitative Plant Science Initiative at Brookhaven National Laboratory.

References

Al-Azzawi, A., Ouadou, A., Tanner, J. J. & Cheng, J. (2019). *Genes*, **10**, 666.
Banjac, J., Donati, L. & Defferrard, M. (2021). *arXiv*:2104.06237.
Bartesaghi, A., Merk, A., Banerjee, S., Matthies, D., Wu, X., Milne, J. L. & Subramaniam, S. (2015). *Science*, **348**, 1147–1151.
Bendory, T., Bartesaghi, A. & Singer, A. (2020). *IEEE Signal Process. Mag.* **37**, 58–76.

electronic reprint

Bepler, T., Kelley, K., Noble, A. J. & Berger, B. (2020). *Nat. Commun.* **11**, 5208.

Bepler, T., Morin, A., Rapp, M., Brasch, J., Shapiro, L., Noble, A. J. & Berger, B. (2019). *Nat. Methods*, **16**, 1153–1160.

Bepler, T., Zhong, E. D., Kelley, K., Brignole, E. & Berger, B. (2019). *arXiv*:1909.11663.

Bibas, K., Weiss-Dicker, G., Cohen, D., Cahan, N. & Greenspan, H. (2021). *arXiv*:2101.03549.

Bracewell, R. N. (1956). *Aust. J. Phys.* **9**, 198–217.

Campbell, M. G., Veesler, D., Cheng, A., Potter, C. S. & Carragher, B. (2015). *eLife*, **4**, e06380.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009). *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Piscataway: IEEE.

Frank, J. & Al-Ali, L. (1975). *Nature*, **256**, 376–379.

Greenberg, I. & Shkolnisky, Y. (2017). *J. Struct. Biol.* **200**, 106–117.

Gupta, H., McCann, M. T., Donati, L. & Unser, M. (2021). *IEEE Trans. Comput. Imaging*, **7**, 759–774.

Gupta, H., Phan, T. H., Yoo, J. & Unser, M. (2020). *Computer Vision – ECCV 2020 Workshops*, edited by A. Bartoli & A. Fusiello, pp. 429–444. Cham: Springer.

Huang, Q., Zhou, Y., Du, X., Chen, R., Wang, J., Rudin, C. & Bartesaghi, A. (2020). *arXiv*:2011.11020.

Jiménez-Moreno, A., Střelák, D., Filipovič, J., Carazo, J. & Sorzano, C. (2021). *J. Struct. Biol.* **213**, 107712.

Kehl, W., Manhardt, F., Tombari, F., Ilic, S. & Navab, N. (2017). *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1521–1529. Piscataway: IEEE.

Kingma, D. P. & Ba, J. (2014). *arXiv*:1412.6980.

Labbé, Y., Carpentier, J., Aubry, M. & Sivic, J. (2020). *Computer Vision – ECCV 2020*, edited by A. Bartoli, H. Bischof, T. Brox & J.-M. Frahm, pp, 574–591. Cham: Springer.

Li, H., Zhang, H., Wan, X., Yang, Z., Li, C., Li, J., Han, R., Zhu, P. & Zhang, F. (2021). *bioRxiv*, doi:2021.05.10.443396.

Manhardt, F., Arroyo, D. M., Rupprecht, C., Busam, B., Birdal, T., Navab, N. & Tombari, F. (2019). *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6841–6850. Piscataway: IEEE.

McSweeney, D. M., McSweeney, S. M. & Liu, Q. (2020). *IUCrJ*, **7**, 719–727.

Miolane, N., Poitevin, F., Li, Y.-T. & Holmes, S. (2020). *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 970–971. Piscataway: IEEE.

Murray, R. M., Li, Z. & Sastry, S. S. (2017). *A Mathematical Introduction to Robotic Manipulation.* Boca Raton: CRC Press.

Palovcak, E., Asarnow, D., Campbell, M. G., Yu, Z. & Cheng, Y. (2020). *IUCrJ*, **7**, 1142–1150.

Park, K., Patten, T. & Vincze, M. (2019). *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7668–7677. Piscataway: IEEE.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. & Lerer, A. (2017). *31st Conference on Neural Information Processing Systems (NIPS 2017)*.

Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C. & Ferrin, T. E. (2004). *J. Comput. Chem.* **25**, 1605–1612.

Punjani, A., Rubinstein, J. L., Fleet, D. J. & Brubaker, M. A. (2017). *Nat. Methods*, **14**, 290–296.

Rad, M. & Lepetit, V. (2017). *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3848–3856. Piscataway: IEEE.

Scheres, S. H. W. (2012*a*). *J. Mol. Biol.* **415**, 406–418.

Scheres, S. H. W. (2012*b*). *J. Struct. Biol.* **180**, 519–530.

Simonyan, K. & Zisserman, A. (2014). *arXiv:*1409.1556.

Vainshtein, B. K. & Goncharov, A. B. (1986). *Sov. Phys. Dokl.* **31**, 278.

Van Heel, M. (1987). *Ultramicroscopy*, **21**, 111–123.

Wagner, T., Merino, F., Stabrin, M., Moriya, T., Antoni, C., Apelbaum, A., Hagel, P., Sitsel, O., Raisch, T., Prumbaum, D., Quentin, D., Roderer, D., Tacke, S., Siebolds, B., Schubert, E., Shaikh, T. R., Lill, P., Gatsogiannis, C. & Raunser, S. (2019). *Commun. Biol.* **2**, 218.

Wang, F., Gong, H., Liu, G., Li, M., Yan, C., Xia, T., Li, X. & Zeng, J. (2016). *J. Struct. Biol.* **195**, 325–336.

Wang, L., Singer, A. & Wen, Z. (2013). *SIAM J. Imaging Sci.* **6**, 2450–2483.

Wong, W., Bai, X.-C., Brown, A., Fernandez, I. S., Hanssen, E., Condron, M., Tan, Y. H., Baum, J. & Scheres, S. H. W. (2014). *eLife*, **3**, e03080.

Xie, R., Chen, Y.-X., Cai, J.-M., Yang, Y. & Shen, H.-B. (2020). *J. Chem. Inf. Model.* **60**, 2614–2625.

Zhong, E. D., Bepler, T., Berger, B. & Davis, J. H. (2021). *Nat. Methods*, **18**, 176–185.

Zhong, E. D., Bepler, T., Davis, J. H. & Berger, B. (2019). *arXiv:*1909.05215.

Zhu, Y., Ouyang, Q. & Mao, Y. (2017). *BMC Bioinformatics*, **18**, 48.

electronic reprint