Cryo2SSNet: 2-Stage Search Supervised Network for Cryo-electron Microscopy Single-Particle Reconstruction

Bingliang Zhang and Zitian Tang

IIIS, Tsinghua University, China
{zhangbl19, tzt19}@mails.tsinghua.edu.cn

Abstract. Cryogenic electron microscopy (cryo-EM) single-particle reconstruction is a powerful technique to elucidate the 3D structure of proteins and other biomolecules from a set of noisy and arbitrarily oriented 2D projection images. This paper presents *2-stage Search Supervised Network* (Cryo2SSNet), an auto-encoder architecture enhanced by a beam search module to jointly infer orientations and reconstruct a 3D density map from a random initialization in a 2-stage pipeline. In the framework, the encoder is to predict orientations for individual images while the decoder is to simulate the cryo-EM imaging process by directly maintaining the 3D density map of the target biomolecule. To stabilize predicted orientations, we propose *denoised heuristic beam search* (DHBS) to provide an explicit weak-supervision for encoder and refined orientations for reconstruction. We evaluate our method on several simulated protein datasets with different signal-to-noise ratios (SNR) and demonstrate the high precision for 3D density map reconstruction.

Keywords: Cryo-EM, Tomographic reconstruction, Auto-encoder

1 Introduction

Biomolecules perform most of the activities in living cells. As the function of a biomolecule is highly correlated to its structure, in order to investigate its activity, it is a key step to determine its structure. The most commonly used ways to do this include X-ray crystallography, nuclear magnetic resonance, and cryogenic electron microscopy (cryo-EM) [19]. With the long-term development, cryo-EM gains the ability to determine biomolecular structures at near-atomic resolution [12]. Though the structure resolution derived by cryo-EM cannot exceed that of X-ray crystallography, combining the cryo-EM structure with the chemical structure of the biomolecule, it is able to produce an informative atomic model or even directly locate every individual atom [19].

To determine the structure of a biomolecule using cryo-EM, about $10^{4\sim7}$ copies of this biomolecule are ice-embedded to a thin vitreous slice. Cryo-EM imaging is then performed over these randomly oriented particles and results in a micrograph, containing the projections of each particle. These projections are

picked from the micrograph and represented as equal-size square images. The projection of a particle can be approximated by an integral of its electron density map, convoluted with the point-spread function (PSF) of the cryo-EM. The final step is to reconstruct the density map based on these projection images, known as tomographic reconstruction. In this procedure, each derived image is of a single distinct particle, hence this process is called single-particle reconstruction (SPR). This allows cryo-EM to capture the heterogeneous conformations of the biomolecule, making cryo-EM advantageous over others. But in many works, including this one, a unique conformation assumption is made to simplify the problem.

The unknown particle poses and extremely strong noises make the tomographic reconstruction challenging. For each projection image, the orientation of the captured particle is unknown, and there might be an unknown translational shift of particle center due to the inaccuracy of particle picking. Besides, the major part of the signal detected by cryo-EM is from the surrounding ice and other noisy sources, rather than the particles. This causes an extremely low signal-to-noise ratio (SNR) of the captured images.

Many approaches have been proposed to solve the tomographic reconstruction problem, including classical ones used in current cryo-EM software packages, and those using deep learning emerging in recent years. In this work, a new deep-learning-based method, *2-stage search supervised network* (Cryo2SSNet), is developed towards tomographic reconstruction in cryo-EM. Specifically, it is an auto-encoder-based model with a search module, whose encoder is a neural network inferring the particle poses, and decoder is a cryo-EM simulator. To balance pose estimation accuracy and efficiency in strongly noisy conditions, we propose *denoised heuristic beam search* (DHBS) strategy as a search module. Besides, our method adopts a 2-stage training pipeline, in the first of which the encoder is weak-supervised by the search module and in the second of which the model is trained in an end-to-end manner. Cryo2SSNet is tested on various simulated protein datasets with low SNR and outperforms previous deep-learning-based methods in reconstruction resolution.

2 Related Works

The cryo-EM SPR problem has been explored since decades ago. While most current cryo-EM software packages are using classical methods like expectation-maximization (EM) algorithm, some deep learning methods are proposed recently to tackle this problem. These two categories are introduced here respectively. Besides, projection-matching, a technique used in a class of methods, which is improved in this work, is discussed in details.

2.1 Current Methods

A major factor making tomographic reconstruction hard is the unknown particle poses. Once the poses are provided, the density map can be reconstructed

by maximum-likelihood (ML) estimation accurately. Hence a straightforward and classical solution is to estimate the particle poses, among which projection-matching-based methods [14,15,1] are popular. Pose estimation and map refinement are performed iteratively. However, the extremely low SNR makes the estimation inaccurate, leading to bad reconstruction.

Instead of finding a certain pose, another class of methods [17,16] maintains a pose distribution for each particle and performs maximum-a-posteriori (MAP) optimization. EM algorithm is used to solve this problem, where E-step updates the pose distribution and M-step updates the density map. These two steps are performed iteratively until convergence. CryoSPARC [16] is one of the most widely used softwares. It proposes to solve the MAP via stochastic gradient descent (SGD) to derive a solution *ab initio*, and performs a subsequent refinement.

2.2 Deep Learning for Cryo-EM

Deep learning, having achieved state-of-the-art performances in a wide range of computer vision tasks, is already introduced into the cryo-EM SPR process, such as being used to denoise the micrographs [2] and to pick particles from the micrographs [23,26,21,22,3].

Works applying deep learning to tomographic reconstruction are proposed in recent years. [9] uses convolutional neural networks (CNNs) to refine pose estimation so that a more accurate density map can be reconstructed. CryoPoseNet [13] utilizes an auto-encoder framework to address the whole reconstruction problem, in which a CNN-based encoder learns to infer the pose of each particle, and a decoder maintains a density map and simulates the cryo-EM imaging process. A reconstruction loss calculated between the dataset images and the simulated reconstructed images so that the framework is trained in an end-to-end manner. CryoGAN [6] uses a paradigm of generative adversarial networks (GANs) [4] to sidestep pose estimation, but the price is that the reconstruction resolution is relatively low.

Some works aim to model the continuous heterogeneous conformation of a biomolecule [7,25]. [7] is a modification of CryoGAN, while [25] uses variational auto-encoder (VAE) [11] to infer a latent conformation representation for each particle.

2.3 Projection-Matching for Cryo-EM

Projection-matching is to find the particle pose via comparing the projections with dataset images. In basic projection-matching-based methods [14,15], the possible poses are discretized and grid search is performed. [5,17] use local search strategy to reduce computational cost, where grid search is only performed in a region near previous estimation, similar to our heuristic search strategy. Even though, grid-search-based projection-matching is still much time-consuming [18].

In [25], branch and bound (BNB) search is proposed. The search space is subdivided hierarchically and the branches with high matching loss are pruned.

Our beam search uses the same subdivision method but different pruning and stopping criterion.

3 Problem Statement

Tomographic reconstruction aims to reconstruct the electron density map $\phi \in \mathbb{R}^{n\times n\times n}$ from a given set of projection images $\{I_1, I_2, \cdots, I_N\}$, where n is the box size of the map, equal to the side length of given images, and N is the dataset size. Since each $I_i \in \mathbb{R}^{n\times n}$ is derived from the cryo-EM imaging process, it can be approximately formulated as

$$I_i = C_{d_i} \otimes P_z \circ (R_i \circ \phi) + \epsilon_i, \tag{1}$$

where $\epsilon_i \in \mathbb{R}^{n\times n}$ is an additive noise, and C_{d_i} is convolutional coefficients called point-spread function (PSF), which is determined by the contrast transfer function (CTF) parameters d_i of the cryo-EM. R_i here is a rotation operator, rotating the map ϕ according to the underlying rotation matrix $r_i \in SO(3)$ of the i-th particle's orientation. P_z is a projection operator along z-axis, defined as

$$(P_z \circ \phi)_{x,y} = \int_{-\infty}^{+\infty} \phi_{x,y,z} \mathrm{d}z. \tag{2}$$

Once the density map ϕ is represented as a 3-dimensional array, the integral in Eq. (2) can be approximated by summation.

Notice that the d_i is known parameter for all i, while the ϵ_i and R_i (and the underlying r_i) are unknown here. Besides, we omit the translational shift in particle pose. It can be added into Eq. (1) as a translation operator with unknown parameters if necessary.

4 Method

4.1 Overview

We consider $\Omega \subset \mathbb{R}^{n\times n}$ as the space of normalized 2D noisy projection images, $\tilde{\Omega} \subset \mathbb{R}^{n\times n}$ as the space of normalized noiseless ones, and $SO(3)$ as the special orthogonal group in 3-dimensional space. An auto-encoder architecture is adopted to simulate the imaging process of cryo-EM. Specifically, the encoder, $E_\theta : \Omega \to SO(3)$, is a convolutional network to predict the pose of an image, and decoder, $D_\phi : SO(3) \to \tilde{\Omega}$, is a differentiable cyro-EM physical simulator. For example, if the orientation operator of the i-th image is R_i together with CTF parameters d_i, the decoding process might be formulated as

$$D_\phi = N(C_{d_i} \otimes P_z \circ (R_i \circ \phi)), \tag{3}$$

where $N(\cdot)$ is normalization operator for 2D spacial pixels, transferring the mean and standard deviation of projection images to 0 and 1, respectively. Notice

that such normalization is effective to blind the numerical differences among various datasets, which further guarantees the generalizability of model hyper-parameters.

Given a set of noisy projection images $\{I_1, \cdots, I_N\} \subset \Omega$, the eltimate goal is to reconstruct the 3D density map ϕ. The simplest way is to train a model in an end-to-end manner by minimizing a reconstruction loss shown in Eq. (4)

$$L(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \ell_{rec}(I_i, (D_\phi \circ E_\theta)(I_i)). \tag{4}$$

However, the trick part here is that orientation is not an intrinsic property of an image. Instead, it depends on the orientation of the 3D density map. This makes encoder hard to learn before it implicitly learns the orientation information of the current density map by gradient information. So, direct end-to-end training would be even impossible to converge at all for complex biomolecules, which is shown in Section 5.2.

To alleviate this issue, we propose *denoised heuristic beam search* (DHBS) as a shortcut for encoder to locate density map orientation by directly learning searched best poses. When encoder adapts to the density map, it is then trained in an end-to-end manner.

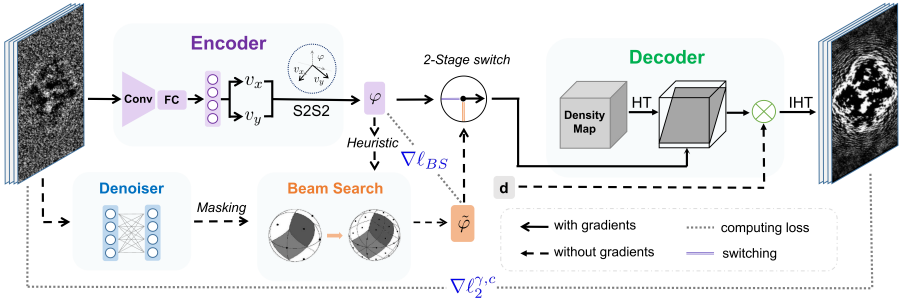


Fig. 1. Overall Cyro2SSNet architecture. Cyro2SSNet is composed of an auto-encoder framework and a search module. In the auto-encoder part, encoder is a CNN performing pose inference for each image, and decoder is a cryo-EM simulator, maintaining a current reconstructed density map and performing imaging process along given poses. The search module looks for a projection orientation matching a given image via DHBS. The training pipeline consists of 2 stages. In the first stage, the searched pose from DHBS supervises encoder and is fed to decoder. In the second stage, DHBS is turned off and the auto-encoder is trained in an end-to-end manner.

4.2 Auto-encoder Architecture

Our overall auto-encoder model consists of two parts, encoder E_θ and decoder D_ϕ. In practice, encoder E_θ takes in a batch of *normalized* noisy projection images and outputs predicted poses (a batch of rotation matrices), while decoder simulates the projection process given poses and outputs *normalized* noiseless projection images. One can refer to Fig. 1, and details are explained below.

Encoder is a convolutional neural network (CNN) followed by a pose prediction head. To output a rotation matrix, we use S2S2 as our parameterization method, which first maps the output feature vector to two 3-dimensional vectors and computes the uniquely determined rotation matrix.

Decoder is a differentiable cryo-EM simulator, modeling the density map as a voxel grid. The simulation is achieved efficiently with the help of the *Fourier slice theorem*, which claims that the Fourier transform of 2D projections is equivalent to the slices of Fourier transformed 3D map. However, complex number induced by Fourier transformation makes gradient backpropagation inconvenient. So we use Hartley transform instead, which holds the slice theorem while eliminats the complex number issue. As shown in Fig. 1, given a batch of poses, 3D Hartley transform is performed on the density map, and slices are extracted according to the poses. The final projection images are obtained by performing inverse Hartley transform on CTF-corrupted slices.

4.3 Denoised Heuristic Beam Search

Due to the highly non-convex property and no-reference problem of encoder, we present *denoised heuristic beam search* (DHBS) to alleviate the optimization difficulty. Instead of directly passing predicted poses from encoder for subsequent decoding, we stabilize and refine the predicted poses by a heuristic search. Specifically, if $r_i \in SO(3)$ is the predicted pose by encoder for the i-th image, DHBS finds $\tilde{r}_i$ defined in Eq. (5), where $De(\cdot)$ denotes a pretrained denoiser and $D_\phi(\cdot)$ denotes decoder. Rather than a global exhaustive search, DHBS is a local heuristic beam search that dramatically reduces the search cost. There are two uses of the output $\tilde{r}_i$: 1) serves as a weak-supervision to r_i and 2) guides reconstruction. The training pipeline with DHBS is described in Algorithm 1 and each component is explained in detail then.

$$\tilde{r}_i \approx \underset{r \in Neighbour(r_i)}{\operatorname{argmin}} \ell_2[De(I_i), De(D_\phi(r))]. \tag{5}$$

Iterative discretization in SO(3). To find $\tilde{r}_i$, the central problem is to model and discretize $Neighbour(r_i)$ in a reasonable manner. As introduced by [24], $SO(3)$ can be discretized iteratively by Hopf fibration. Let $G_k \subset SO(3)$ be a grid in resolution k, whose elements represent the centers of uniformly partitioned regions in $SO(3)$. Each region in resolution k can be further divided into 8 subregions uniformly. All the centers of these subregions compose G_{k+1}, the grid of next resolution level. Concretely, define the operator $SubDiv$ as

$$SubDiv(e; k) = \{8 \text{ elements divide region of } e\} \subset G_{k+1}, \quad \forall e \in G_k, \tag{6}$$

then

$$G_{k+1} = \bigcup_{e \in G_k} SubDiv(e; k). \tag{7}$$

The base grid G_0 has 72 elements due to geometric property of $SO(3)$, so $|G_k| = 72 \times 8^k$. The naive burte force search is to find the pose globally by modeling $Neighbour(r_i) \approx G_k$ for a certain k, i.e.,

$$\tilde{r}_i^{BF} = \underset{r \in G_k}{\operatorname{argmin}} \ell_2[De(I_i), De(D_\phi(r))]. \tag{8}$$

However, the computational cost is exponential with respect to resolution k, hence a wiser search strategy is necessary for efficiency.

Heuristic Beam Search. As mentioned above, the search process should be done efficiently by trading off accuracy and efficiency, so we present *heuristic beam search* to evaluate poses iteratively and achieve *linear* time cost w.r.t. resolution level k. First we select c nearest neighbors of r_i in the base grid G_0. Then these c poses are subdivided into $8c$ poses in the next level by $SubDiv(\cdot)$ operator. Compute the ℓ_2 costs for these poses and keep track of the top-c with minimal costs. This division-evaluation process is repeated for i times and the best pose among the last c poses is returned, termed as $\tilde{r}_i^{BS}$. The search width c and depth i can be tuned for an accuracy-efficiency tradeoff. The beam search degrades to greedy search when $c = 1$ and remains more currently sub-optimal poses when c is larger. Notice the heuristic trick that picking nearest neighbours of r_i instead of comparing all candidates in G_0 reduces nearly half of the time cost in practice.

Gradient Descent Enhancement. Though the searched pose usually results in a smaller cost compared with predicted pose by encoder, there is still a theoretical lower bound of distance between searched pose and the globally best one due to discrete nature of grid for a finite search depth. To tackle this problem, we first make the following assumption.

Assumption 1 (Local Convexity) *The cost landscape is* locally convex *near the globally best pose and the previously searched pose $\tilde{r}_i^{BS}$ is close to globally best pose. So that it is within this convex region.*

There are some empirical evidence for the above assumption in Appendix B.3. Based on this, a further refinement after beam search is performed to update the pose locally and continuously by gradient descent. Specifically, we update the searched pose $\tilde{r}_i^{BS}$ for n steps with learning rate η as formulated in Eq. (9). This gradient descent can be performed since the cost is differentiable with respect to $\tilde{r}_i^{BS}$.

$$\tilde{r}_i^{BS} \leftarrow \tilde{r}_i^{BS} - \eta \cdot \nabla_{\tilde{r}_i^{BS}}[\ell_2[De(I_i), De(D_\phi(\tilde{r}_i^{BS}))]]. \tag{9}$$

Denoiser. Due to the extremely low SNR of images, direct comparison between noisy and noiseless images may cause some issues (e.g. a sub-optimal search result). Hence, we denoise the images while computing the search cost. The denoiser we use is a simple neural network, described in Appendix A.2.

It is noticeable that perfect denoising quality is not required. The denoised images are only utilized to evaluate searched candidates, rather than to be reconstructed. Denoiser helps DHBS to focus on low-frequency features and find right poses in early stage and is turned off after a few epochs. This fits to a coarse-to-fine criterion, proven to be useful in many computer vision tasks.

Composing all the techniques above, an overall algorithm of DHBS is shown in Algorithm 1.

Algorithm 1 Denoised Heuristic Beam Search (DHBS) algorithm

Input: A noisy image I and predicted pose r, denoiser $De(\cdot)$, decoder $D_\phi(\cdot)$ and hyper-parameters (c, i, n, η)
Output: Refined pose $\tilde{r}$

$S_0 = \{e \mid e \in G_0 \text{ and } e \text{ is top } c \text{ nearest neighbour of } r\}$
for $k = 1, 2, \ldots, i$ **do**
  $\tilde{S}_k = \bigcup_{e \in S_{k-1}} SubDiv(e; k-1)$ (sub-division)
  $S_k = \text{low-c}_{e \in \tilde{S}_k} \ell_2[De(I), De(D_\phi(e))]$ (best c candidates)
end for
$\tilde{r} = \text{argmin}_{e \in S_k} \ell_2[De(I), De(D_\phi(e))]$
for $j = 1, 2, \ldots, n$ **do**
  $\tilde{r} \leftarrow \tilde{r} - \eta \cdot \nabla_{\tilde{r}} \ell_2[De(I), De(D_\phi(\tilde{r}))]$
end for
Return: $\tilde{r}$

4.4 2-Stage Switch and Loss Objectives

First, a metric measuring distances in $SO(3)$ space is needed in our loss objectives. Proved in [8], there are a few equivalent metrics, among which *quaternion distance* is the one we use. For $R_1, R_2 \in SO(3)$, let $\varphi_1, \varphi_2 \in \mathbb{R}^4$ be quaternions of R_1, R_2, respectively. The quaternion distance between R_1 and R_2 is

$$d_{quat}(R_1, R_2) = \min(||\varphi_1 - \varphi_2||_2, ||\varphi_1 + \varphi_2||_2). \tag{10}$$

As mentioned before, DHBS helps encoder to implicitly learn the orientation and speeds up reconstruction by polishing predicted poses from encoder. When encoder learns to infer poses close to globally best ones, the optimization difficulty largely mitigates. Based on this fact, we present a 2-stage training pipeline to combine search-supervised stage and end-to-end training stage.

During the first stage, the gradient is not passed from decoder back to encoder. Instead, the searched poses from DHBS are used as weak-supervision for training encoder and passed to decoder. Specifically, if the predicted pose is r and searched pose is $\tilde{r}$, an auxiliary loss for weak-supervision is

$$\ell_{BS}(r, \tilde{r}) = d^2_{quat}(r, \tilde{r}). \tag{11}$$

As for the reconstruction loss, we notice that vanilla ℓ_2 loss is *infeasible* to define distance between space Ω and $\tilde{\Omega}$, since they are both normalized. Instead,

we proposed an unbiased estimation of vanilla ℓ_2 loss in Eq. (12), termed as $\ell_2^{\gamma,c} : \Omega \times \tilde{\Omega} \to \mathbb{R}^+$, where γ, c are obtained according to the SNR of images. Formally, if the SNR of space Ω is known, say s, then $\gamma = \sqrt{1 + 1/s}$ and $c = 1/s$. The x, y in the formula are two-dimensional indices. The derivation is shown in Appendix A.3. Once the SNR of a given dataset is unknown, we find a slightly overestimated SNR can make our model work as well, which makes it more flexible.

$$\ell_2^{\gamma,c}(I_i, \tilde{I}_i) = \frac{1}{n^2} \sum_{1 \le x,y \le n} (\gamma \cdot I_i - \tilde{I}_i)_{x,y}^2 - c, \quad I_i \in \Omega, \quad \tilde{I}_i \in \tilde{\Omega}. \tag{12}$$

We notice that passing searched pose $\tilde{r}$ to decoder is better than passing predicted pose r from encoder, since $\tilde{r}$ is closer to the globally best pose hence better for reconstruction. By doing so, the gradient is cut between encoder and decoder, and $\ell_{BS}, \ell_2^{\gamma,c}$ optimize encoder and decoder respectively. They are combined by a tunable hyper-parameter, resulting in the loss function in the first stage, as

$$L_{S1}(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \ell_2^{\gamma,c}(I_i, D_\phi(\tilde{r}_i)) + \lambda \cdot \ell_{BS}(r_i, \tilde{r}_i), \tag{13}$$

$$\textbf{where: } r_i = E_\theta(I_i),\ \tilde{r}_i = DHBS(r_i; D_\phi, I_i).$$

During the second stage, we assume encoder has already implicitly learned a rough orientation of 3D density map. Thus, encoder and decoder can be fine-tuned in an end-to-end manner with loss objective

$$L_{S2}(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \ell_2^{\gamma,c}(I_i, D_\phi(E_\theta(I_i))) \tag{14}$$

We name our method *2-stage Search Supervised Network* (Cryo2SSNet). The overall algorithm pipeline of Cryo2SSNet is summarized in Algorithm 2.

5 Experimental Results

We now describe our experiments and results on several simulated protein datasets, including 4AKE, 5A1A, 7ACT and 7AD1. We derived the atomic models in PDB format from protein data bank[1] and generated their density maps using the open-source software EMAN2 [20], and built simulated datasets by CryoSPARC [16]. All simulated datasets use the default CTF parameters in CryoSPARC.

In each experiment, we reconstructed the density map with full dataset and calculated the FSC curve between the density map reconstruction and the ground truth. The resolution cutoff in FSC curve is 0.5.

[1] https://www.rcsb.org/

Algorithm 2 2-Stage Search Supervised Network (Cryo2SSNet)

Input: Noisy projection image dataset $\{I_1, I_2, \cdots, I_N\}$, 2-stage epochs (E_1, E_2).
Output: Reconstructed density map ϕ

Initialize E_θ with random weight and D_ϕ with scaled uniform random distribution $U(0, 0.1)$ in region within an estimated protein radius and zero outside.
# Stage 1
for $epoch = 1, 2, \ldots, E_1$ **do**
  for $\mathbf{I} \leftarrow$ sampled batch of images **do**
    $\mathbf{r} \leftarrow E_\theta(\mathbf{I})$
    $\tilde{\mathbf{r}} \leftarrow DHBS(\mathbf{r}; D_\phi, \mathbf{I})$
    $L = \ell_2^{\gamma,c}(\mathbf{I}, D_\phi(\tilde{\mathbf{r}})) + \lambda \cdot \ell_{BS}(\mathbf{r}, \tilde{\mathbf{r}})$
    Update θ, ϕ with $\nabla_{\theta,\phi} L$
  end for
end for
# Stage 2
for $epoch = E_1 + 1, \ldots, E_1 + E_2$ **do**
  for $I \leftarrow$ sampled batch of images **do**
    $\mathbf{r} \leftarrow E_\theta(\mathbf{I})$
    $L = \ell_2^{\gamma,c}(\mathbf{I}, D_\phi(\mathbf{r}))$
    Update θ, ϕ with $\nabla_{\theta,\phi} L$
  end for
end for
Return: ϕ

5.1 Reconstruction

Experiment Settings For 4AKE, the protein used in CryoPoseNet [13], we generated a 1.6 Å density map of size $128 \times 128 \times 128$, with voxel size 0.8 Å, which corresponds to a Nyquist-Shannon limit of 1.6 Å for reconstruction. We created 1 noiseless dataset and 3 noisy datasets with SNRs equal to 10 dB, 0 dB, −10 dB, respectively. Each dataset contains $10,000$ projection images.

For 5A1A, the one used in CryoGAN [6], following their experimental settings, we generated a 2.5 Å density map of size $256 \times 256 \times 256$, with voxel size 1.25 Å, corresponding to a Nyquist-Shannon limit of 2.50 Å. 3 datasets with SNRs equal to −5.2 dB, −10 dB, −20 dB, respectively, were created. Each dataset contains $40,000$ images. Though 5A1A is in a structure of D2 symmetry, unlike [6], we do not explicitly utilize this property in our reconstruction method.

One can refer to Appendix B for detailed settings and additional results for the remaining proteins.

In each experiment, we ran Cryo2SSNet for 7500 steps in total with a batch size of 32, which corresponds to 6 epochs for 5A1A and 24 epochs for others. We divided the total number of epochs equally for two stages. As for DHBS, the search depth and search width were set to be 5 and 4, respectively. More detailed training parameters are listed in Appendix B.2.

Results Cryo2SSNet achieved a reconstruction resolution of 2.20 Å for 4AKE with SNR=−10 dB and 4.64 Å for 5A1A with SNR=−20 dB. Our reconstruc-

tion results are shown in Table 1. It demonstrates that our proposed method reconstructs the density maps of 4AKE and 5A1A with fine qualities close to the Nyquist-Shannon limit, and it can generalize to different noise levels and protein types. Visualization of our reconstruction results is shown in Fig. 2.

Table 1. The reconstruction resolutions of Cryo2SSNet. The results are compared with ones reported by previous proposed methods.

Protein	SNR (dB)	Resolution (Å)	Previous Resolution(Å)
4AKE	∞	1.74	/
	10	1.99	/
	0	1.94	2.57 (CryoPoseNet)[2]
	-10	2.20	/
5A1A	-5.2	2.68	8.30 (CryoGAN)
	-10	3.14	/
	-20	4.64	15.30 (CryoGAN)

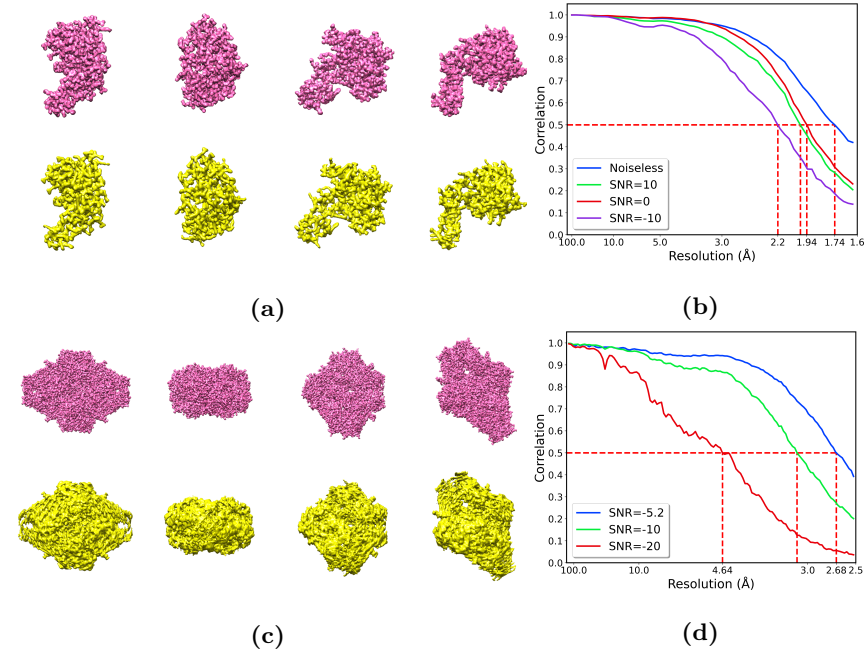


Fig. 2. Reconstruction results. (a) and (c) visualize our reconstructed density maps for 4AKE with SNR=−10 dB and 5A1A with SNR=−20 dB, respectively. The pink ones are ground truths while the yellow ones are reconstructions. (b) and (d) shows the FSC curves in various noise levels for 4AKE and 5A1A, respectively.

[2] The original paper of CryoPoseNet didn't mention the resolution of density map used to simulate dataset. So we compare it with the one of same SNR and use as reference only.

We notice that the reconstructed density map converges quickly during the first several epochs as illustrated in Fig. 4. It's indicated that the density map converges to a rough sketch first and is refined in details gradually. This also reveals that constructing a rough initial density map is much easier compared to structure refinement. The challenging part of tomographic reconstruction is, in order to derive a high-resolution reconstruction, an *accurate enough* pose for each particle is required to be inferred according to the extremely noisy images. After our first stage, the density map is almost converged. The second stage coordinates encoder and decoder by training them jointly at a fine pace. By utilizing DHBS to enhance supervision for encoder, we observe a high convergence speed as shown in Fig. 3, which is necessary for the second training stage.

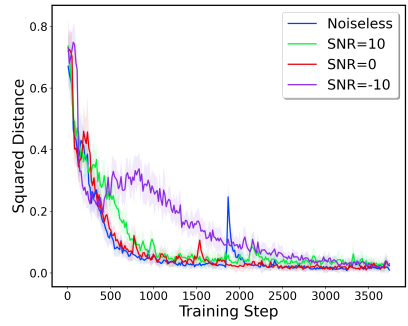


Fig. 3. Curves of squared distance between predicted and searched poses, tested on 4AKE.

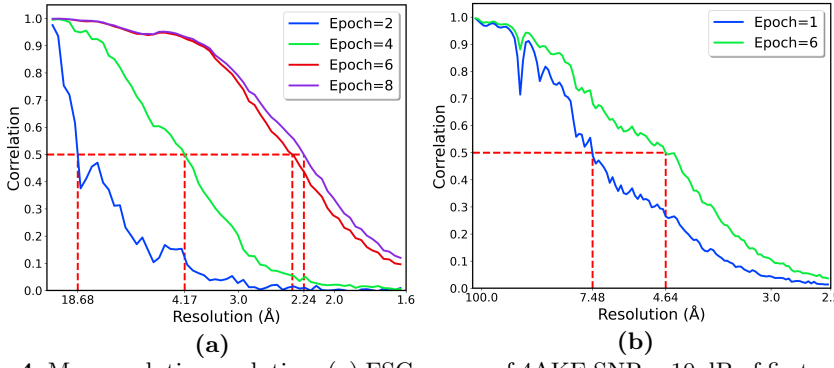


Fig. 4. Map resolution evolution. (a) FSC curves of 4AKE SNR=-10 dB of first several epochs. (b) Comparison of starting epochs and last epochs of 5A1A SNR=-20 dB.

5.2 Necessity of DHBS

It can be verified that DHBS is necessary for extremely noisy dataset, e.g., 5A1A with SNR=−20 dB. We run reconstruction by eliminating stage 1 but run twice of the total epochs (12 epochs) for time compensation. We notice, in this setting, the density map cannot converge at all. The comparison of reconstruction results is shown in Fig. 5. By eliminating DHBS, the result of end-to-end training is even worse than the first epoch of Cryo2SSNet,

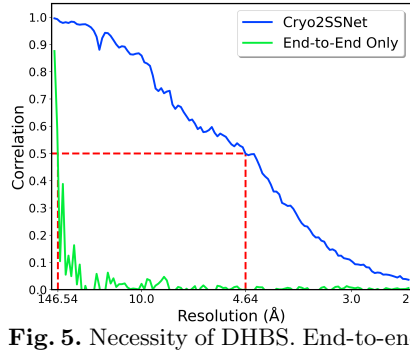


Fig. 5. Necessity of DHBS. End-to-end training does not work at all in 5A1A SNR=−20 dB.

referring to Fig. 4(b), which indicates that it fails to find a rough sketch of density map.

There are several possible reasons for this. 1) Strong noise in images confuses encoder to implicitly coordinate with decoder, 2) high variance of encoder predictions causes density map fluctuating, and 3) the whole model gets stuck in a bad local minimum. With the help of DHBS, these issues may be mitigated to a degree.

5.3 Effectiveness of Gradient Descent Enhancement

In this part we conduct experiments to verify the effectiveness of gradient descent (GD) enhancement. 5A1A with SNR=−10 dB is selected as testbed. We run our method with and without GD enhancement, keeping other settings the same as described in Appendix B. Also, to further reveal the potential of GD enhancement, we perform reconstruction using Adam [10] instead of vanilla GD in Cryo2SSNet, to update searched pose $\tilde{r}^{BS}$ in each step. The results are shown in Fig. 6, where the curve "w/ GD" is the result of Cryo2SSNet. It is shown that gradient descent enhancement is necessary for Cryo2SSNet to achieve high resolution without enlarging search depth. As indicated by results, GD enhancement with Adam outperforms the vanilla one. One can substitute GD in Cryo2SSNet with Adam for better results, while GD is used throughout this paper for simplicity.

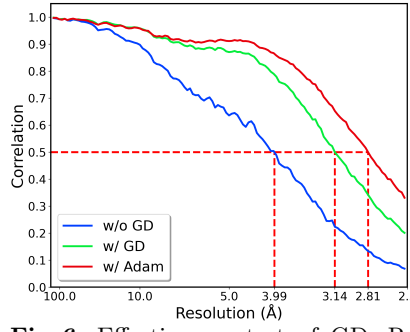


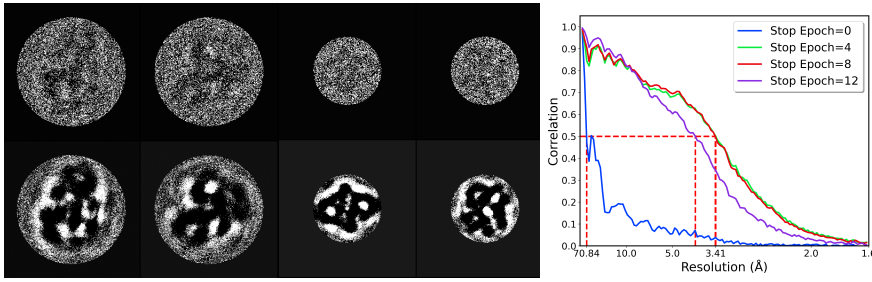
Fig. 6. Effectiveness test of GD. Removing GD enhancement destroys the performance.

5.4 Effectiveness of Denoiser

As we mention in Section 4.3, the denoiser allows a coarse-to-fine process when the images are extremely noisy. The denoising results are visualized in Fig. 7(a). As we can tell, denoiser doesn't remove all the noises, but does emphasize and expose a rough structure of the particle in a certain view, which helps a lot in looking for correct poses in the early stage. In practice, instead of denoising the entire image, we first take a circular mask of a rough particle radius to filter out the noises outside. This radius can be easily estimated by taking numerical average over the entire dataset. Applying this mask helps denoiser to focus on the region where the particle locates.

Additionally, we conduct quantitative experiments using protein 7AD1 with SNR=−10 dB, whose structure is the most complicated among the 4 proteins. Details of this dataset are described in Appendix B.1. To highlight the effect of denoiser, we use search width of 3 and search depth of 3, instead of 4 and 5 in other experiments, and keep other parameters unchanged. In this experiment, we control the number of epochs applying denoiser. The results are shown in

Fig. 7(b). Since there are 12 epochs in the first stage, "Stop Epoch=0" (in short as $D0$ later) means not to apply denoiser and "Stop Epoch=12" means to apply denoiser through the entire first stage. As shown, $D0$ fails completely and $D12$ degrades the performance compared to $D4$ and $D8$. This result coincides with our expectation: denoiser helps to search right poses, but, as a compromise, loses high-frequency information. Therefore, applying denoiser in early stage and turning it off later mimics a coarse-to-fine paradigm in computer vision.



(a) **(b)**

Fig. 7. Qualitative and quantitative analysis of denoiser effectiveness. (a) Denoising results. The first row is the raw images of 7AD1 (first two; SNR=−10 dB) and 5A1A (last two; SNR=−20 dB), while the second row is the denoising results. Denoiser focuses on a particle-size centered region. (b) FSC curves of 7AD1 reconstruction with SNR=−10 dB. Applying denoiser in early epochs and turning it off outperforms other settings.

6 Conclusion

In this paper, we present Cryo2SSNet, a search supervised auto-encoder tackling cryo-EM single-particle reconstruction problem. Specifically, a DHBS method is proposed to help pose inference. The search module is used to supervise the encoder in early stage and turned off later, resulting in a 2-stage training pipeline.

Cryo2SSNet is shown to be able to achieve higher reconstruction resolution than previous deep-learning-based methods on a variety of simulated protein datasets, within comparable time cost. Our experiments indicate that DHBS helps encoder learn to infer poses more accurately, which is necessary to reconstruct high-resolution density maps. Also, we claim that Cryo2SSNet can generalize to different datasets, and can be further tuned for a tradeoff between accuracy and efficiency.

However, there are still limitations of our work. The second stage of Cryo2SSNet can make only minor refinement of the density map, which doesn't meet our expectations. Besides, omission of translational shift in particle pose makes it hard to adapt to real datasets. We leave solving them as future works.

References

1. Baker, T.S., Cheng, R.H.: A model-based approach for determining orientations of biological macromolecules imaged by cryoelectron microscopy. Journal of structural biology **116**(1), 120–130 (1996)
2. Bepler, T., Kelley, K., Noble, A.J., Berger, B.: Topaz-denoise: general deep denoising models for cryoem and cryoet. Nature communications **11**(1), 1–12 (2020)
3. Bepler, T., Morin, A., Rapp, M., Brasch, J., Shapiro, L., Noble, A.J., Berger, B.: Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. Nature methods **16**(11), 1153–1160 (2019)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
5. Grigorieff, N.: Frealign: high-resolution refinement of single particle structures. Journal of structural biology **157**(1), 117–125 (2007)
6. Gupta, H., McCann, M.T., Donati, L., Unser, M.: Cryogan: A new reconstruction paradigm for single-particle cryo-em via deep adversarial learning. IEEE Transactions on Computational Imaging **7**, 759–774 (2021). https://doi.org/10.1109/TCI.2021.3096491
7. Gupta, H., Phan, T.H., Yoo, J., Unser, M.: Multi-cryogan: Reconstruction of continuous conformations in cryo-em using generative adversarial networks. In: European Conference on Computer Vision. pp. 429–444. Springer (2020)
8. Huynh, D.Q.: Metrics for 3d rotations: Comparison and analysis. Journal of Mathematical Imaging and Vision **35**(2), 155–164 (2009)
9. Jiménez-Moreno, A., Střelák, D., Filipovič, J., Carazo, J., Sorzano, C.: Deepalign, a 3d alignment method based on regionalized deep learning for cryo-em. Journal of Structural Biology **213**(2), 107712 (2021)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
12. Kühlbrandt, W.: The resolution revolution. Science **343**(6178), 1443–1444 (2014)
13. Nashed, Y.S.G., Poitevin, F., Gupta, H., Woollard, G., Kagan, M., Yoon, C.H., Ratner, D.: 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 (ICCV) Workshops. pp. 4066–4076 (October 2021)
14. Penczek, P., Radermacher, M., Frank, J.: Three-dimensional reconstruction of single particles embedded in ice. Ultramicroscopy **40**(1), 33–53 (1992)
15. Penczek, P.A., Grassucci, R.A., Frank, J.: The ribosome at improved resolution: new techniques for merging and orientation refinement in 3d cryo-electron microscopy of biological particles. Ultramicroscopy **53**(3), 251–270 (1994)
16. Punjani, A., Rubinstein, J.L., Fleet, D.J., Brubaker, M.A.: cryosparc: algorithms for rapid unsupervised cryo-em structure determination. Nature methods **14**(3), 290–296 (2017)
17. Scheres, S.H.: Relion: implementation of a bayesian approach to cryo-em structure determination. Journal of structural biology **180**(3), 519–530 (2012)
18. Sigworth, F.J.: Principles of cryo-em single-particle image processing. Microscopy **65**(1), 57–67 (2016)

19. Singer, A., Sigworth, F.J.: Computational methods for single-particle electron cryomicroscopy. Annual Review of Biomedical Data Science **3**, 163–190 (2020)
20. Tang, G., Peng, L., Baldwin, P.R., Mann, D.S., Jiang, W., Rees, I., Ludtke, S.J.: Eman2: an extensible image processing suite for electron microscopy. Journal of structural biology **157**(1), 38–46 (2007)
21. Tegunov, D., Cramer, P.: Real-time cryo-em data pre-processing with warp. BioRxiv p. 338558 (2018)
22. Wagner, T., Merino, F., Stabrin, M., Moriya, T., Antoni, C., Apelbaum, A., Hagel, P., Sitsel, O., Raisch, T., Prumbaum, D., et al.: Sphire-cryolo is a fast and accurate fully automated particle picker for cryo-em. Communications biology **2**(1), 1–13 (2019)
23. Wang, F., Gong, H., Liu, G., Li, M., Yan, C., Xia, T., Li, X., Zeng, J.: Deeppicker: A deep learning approach for fully automated particle picking in cryo-em. Journal of Structural Biology **195**(3), 325–336 (2016)
24. Yershova, A., Jain, S., Lavalle, S.M., Mitchell, J.C.: Generating uniform incremental grids on so (3) using the hopf fibration. The International journal of robotics research **29**(7), 801–812 (2010)
25. Zhong, E.D., Bepler, T., Davis, J.H., Berger, B.: Reconstructing continuous distributions of 3d protein structure from cryo-em images. In: International Conference on Learning Representations (2019)
26. Zhu, Y., Ouyang, Q., Mao, Y.: A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. BMC bioinformatics **18**(1), 1–10 (2017)

A Appendix-Method

A.1 Encoder Architecture

Here we describe the architecture used in encoder. The CNN is composed of 3 convolutional blocks followed by 2 fully connected layers (MLP) and a pose prediction head. Each convolutional block consists of 2 Conv-ReLU modules and 1 max-pooling layer. The numbers of channels for each block are $32, 64, 128$, respectively. Each block down-samples feature map by a factor of 2. The hidden dimensions for 2 subsequent fully connected layers are both 512.

The pose is then predicted by S2S2 parameterization. Specifically, two 3D dimensional vectors v_x and v_y are predicted from obtained feature vector through two fully-connected layers. In S2S2, v_y is projected to orthogonal direction of v_x. Then v_x and projected v_y are normalized, termed as $\hat{v}_x$ and $\hat{v}_y$. Then $\hat{v}_z$ is defined as the cross product of $\hat{v}_x$ and $\hat{v}_z$. Finally $\left[\hat{v}_x\ \hat{v}_y\ \hat{v}_z\right]$ is the predicted rotation matrix.

A.2 Denoiser

The denoiser used in this work is a simple MLP, pretrained by the normalized noisy projection images unsupervisedly. It is a 5-layer auto-encoder model. The first layer down-samples the image's dimension (i.e. the total number of pixels) by a factor of f, a tunable hyper-parameter. The 3 consequent hidden layers keep the dimension and the last one recovers the original image size. The denoiser is optimized to minimize the ℓ_2 reconstruction loss.

In the MLP, down-sampling serves as an information bottleneck filtering amount of noise out. CNN is not used here since MLP is capable enough to achieve reasonably good denoising result compared with CNN. Hence we only use an MLP denoiser for simplicity and efficiency.

The denoiser is trained for 7500 steps with batch size of 64, using Adam optimizer with initial learning rate of $1e{-}4$. We use unbiased ℓ_2 loss proposed in Eq. (12) as the optimization objective. Training time for denoiser is negligible compared to reconstruction time (less than 5%). The downsample factors f for each protein are shown in Table 3.

A.3 Unbiased Estimation of ℓ_2 Loss

Here we give the derivation of unbiased estimation of ℓ_2 loss. We represent noiseless projection image (corrupted by CTF already) as a random variable defined in $\mathbb{R}^{n\times n}$, where n is the edge length of image. Besides, we model the noise as a zero-mean random variable. We assume SNR of the images is known. Specifically, let $d, e \in \mathbb{R}^{n\times n}$ be random variables for projection image and noise, following a joint distribution $p(d, e)$. A noisy image in dataset is $x = d + e$. Let $f(\cdot)$ denote our auto-encoder model. $\mathbb{E}(\cdot)$ stands for the expectation of a certain random variable.

Definition 1 (ℓ_2 **Loss).** *For a noiseless image d and noise e, the expected squared error between $f(x)$ and d is ℓ_2 loss shown in Eq. (15).*

$$\ell_2(d, f(x)) = \mathbb{E}_{d,e\sim p(d,e)}\left[\frac{1}{n^2}\sum_{1\le i,j\le n}(d - f(x))_{i,j}^2\right] \tag{15}$$

The expectation is taken over the joint probability distribution $p(d, e)$. We do not explicitly write this later. One problem here is that we would like to perform reconstruction over a normalized output for generalization. Here we give the exact definition for spatial normalization.

Definition 2 (Spacial Normalization). *Given an instance $y \in \mathbb{R}^{n\times n}$, we define $M(y), S(y)$ as the mean and standard deviation over n^2 positions, shown in Eq. (16). And normalization is simply $N(y) = (y - M(y))/S(y)$.*

$$M(y) = \frac{1}{n^2}\left(\sum_{1\le i,j\le n} y_{i,j}\right), \quad S(y) = \sqrt{\frac{1}{n^2}\left(\sum_{1\le i,j\le n}(y - M(y))_{i,j}^2\right)} \tag{16}$$

According to the definition above, we know $N(x) \in \Omega$ and $N(f(x)) \in \tilde{\Omega}$. So what we want is to find out an estimation of $\ell_2(N(d), N(f(x)))$, since this is inaccessible because we do not obtain d from the dataset. So we have to estimate the above term with the help of the entire dataset. Unfortunately, this is unachievable for general joint probability distribution $p(d, e)$. The relation between d and e should be considered also.

Specifically, considering noise e and image d, we assume e is *independent* of d. This assumption is natural. Even though this may not be correct in real dataset, it still serves as a good approximation. On this assumption, we claim the following property.

Assumption 2 (Independence) *The image and noise are independent, i.e., $p(d, e) = p(d) \cdot p(e)$. With this property, for any given point-wise function $\psi(\cdot)$, we have*

$$\mathbb{E}_{d,e}[\psi(d) \cdot e] = \mathbb{E}_{d,e}[\psi(d)] \cdot \mathbb{E}_{d,e}[e] = 0. \tag{17}$$

Here point-wise function means applying the same function to each position repeatedly.

Then we give the definition of our proposed unbiased ℓ_2 loss and the main theorem.

Definition 3 (Unbiased L2 Loss). *Given SNR=s, a noisy image x and a model $f(\cdot)$, let $\gamma = \sqrt{1 + 1/s}$ and $c = 1/s$, then we define $\ell_2^{\gamma,c}$ as*

$$\ell_2^{\gamma,c}(N(x), N(f(x))) = \mathbb{E}_{d,e}\left[\frac{1}{n^2}\sum_{1\le i,j\le n}(\gamma \cdot N(x) - N(f(x)))_{i,j}^2 - c\right]. \tag{18}$$

Theorem 1 (Equivalence). *Given the definition above, we have*

$$\ell_2^{\gamma,c}(N(x), N(f(x))) = \ell_2(N(d), N(f(x))). \tag{19}$$

Proof (Equivalence). First according to the assumptions, we know $\mathbb{E}[e_{i,j}] = 0$ for $\forall 1 \le i, j \le n$, and

$$\text{SNR} = s = \mathbb{E}_{d,e}\left[\frac{S^2(d)}{S^2(e)}\right]. \tag{20}$$

Then we have

$$\begin{aligned}
&\ell_2^{\gamma,c}(N(x), N(f(x))) \\
=&\mathbb{E}_{d,e}\left[\frac{1}{n^2}\sum_{1\le i,j\le n}(\gamma\cdot N(x) - N(f(x)))_{i,j}^2 - c\right] \\
=&\frac{1}{n^2}\sum_{1\le i,j\le n}\mathbb{E}_{d,e}\left[\left(N(f(x)) - \gamma\frac{x - M(x)}{S(x)}\right)_{i,j}^2\right] - c \\
=&\frac{1}{n^2}\sum_{1\le i,j\le n}\mathbb{E}_{d,e}\left[\left(N(f(x)) - \gamma\sqrt{\frac{s}{1+s}}\cdot\frac{d+e-M(d)}{S(d)}\right)_{i,j}^2\right] - c \\
=&\frac{1}{n^2}\sum_{1\le i,j\le n}\mathbb{E}_{d,e}\left[\left(N(f(x)) - N(d) - \frac{e}{S(d)}\right)_{i,j}^2\right] - c \\
=&\frac{1}{n^2}\sum_{1\le i,j\le n}\mathbb{E}_{d,e}[(N(f(x)) - N(d))_{i,j}^2] + \mathbb{E}_{d,e}\left[\frac{S^2(e)}{S^2(d)}\right] - c + \\
&\frac{2}{n^2}\sum_{1\le i,j\le n}\mathbb{E}_{d,e}\left[\left(\frac{N(f(x)) - N(d)}{S(d)}\cdot e\right)_{i,j}\right] \\
=&\ell_2(N(d), N(f(x))) + 0 + \\
&\frac{2}{n^2}\sum_{1\le i,j\le n}\mathbb{E}_{d,e}\left[\left(\frac{N(f(x)) - N(d)}{S(d)}\right)_{i,j}\right]\cdot\mathbb{E}[e_{i,j}] \\
=&\ell_2(N(d), N(f(x))).
\end{aligned} \tag{21}$$

The red term is the one we want and the blue term equals zero due to the definition of SNR. The cyan term equals zero due to our independence assumption. Then the proof is done.

One thing to notice is that Eq. (12) is an empirical estimation of unbiased ℓ_2 loss over the dataset. The meaning is clear with the help of the context. The significance of this theorem is that we are able to optimize an unbiased loss objective even without access to noiseless images.

B More Experimental Results

B.1 Remaining Protein Settings

We have mentioned the setting of simulated dataset of 4AKE and 5A1A in Section 5.1. Here we describe the setting of other protein simulated datasets.

As for 7ACT, we generated a 1.0 Å density map of size $128 \times 128 \times 128$ with voxel size 0.5 Å, which corresponds to a Nyquist-Shannon limit of 1.0 Å. Only noiseless and noisy dataset of SNR=-10 dB were considered, where each one contains $10,000$ projection images.

As for 7AD1, density map resolution was 1.6 Å and size was $256 \times 256 \times 256$, with voxel size 0.8 Å. We still only considered noiseless and noisy dataset of SNR=-10 dB with $10,000$ images for each. A summary of 4 protein datasets is in Table 2.

Table 2. This gives the summary of simulated protein dataset. The Nyquist-Shannon limit is the twice of voxel size.

Protein	Map Resolution (Å)	Voxel Size (Å)	Set Size
4AKE	1.6	0.8	10,000
5A1A	2.5	1.25	40,000
7AD1	1.6	0.8	10,000
7ACT	1.0	0.5	10,000

As for CTF parameters, we use the default values in CryoSPARC for all of our simulated datasets. Specifically, the accelerating voltage is 300 kV, the amplitude contrast is 0.07, and the spherical aberration is 2.7 mm. The mean of two defocus is uniformly drawn from 15000 Å to 20000 Å, and the difference between two defocus is uniformly drawn from 100 Å to 500 Å. The defocus angle is uniformly drawn from 0.2 rad to 1.4 rad.

B.2 Training Details

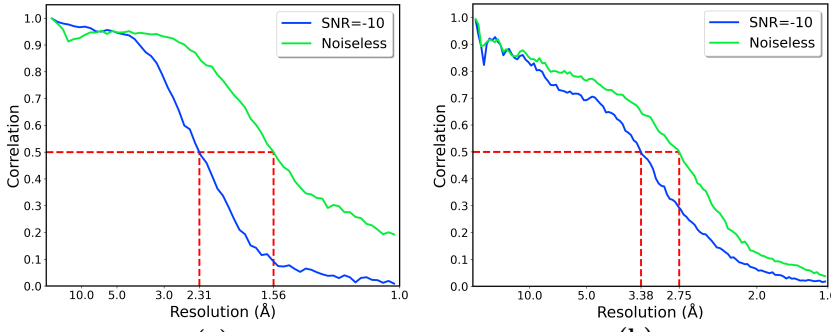
In this section, we describe the training details of our experiments. We use Pytorch for automatic gradient computation. Encoder was initialized by default in Pytorch and the density map in decoder was initialized by a uniform distribution between 0 and 0.1 within a circular region. The outside region was initialized to 0. Protein-sized mask radiuses are listed in Table 3, which stands for the proportion to half-edge length.

The total number of training steps was fixed as 7500, and the batch size is 32. The number of epochs was 6 for 5A1A and 24 for others. The number of epochs in stage 1 and stage 2 were partitioned equally. As a coarse-to-fine process, we replaced denoiser with an identity function after the first 1/3 of stage 1. The search depth and search width were set to be 5 and 4, respectively. Adam was used as the optimizer with initial learning rate of $1e{-}4$ for encoder and $1e{-}2$ for decoder. We only decayed the learning rate of decoder by a factor 10 when

entering stage 2. The balance factor between the two loss terms in stage 1 was set to 1.

As for gradient descent enhancement, we used vanilla GD algorithm for 20 steps with learning rate 0.05.

All the experiments were running on a single NVIDIA Tesla K80 GPU, and the training time is shown in Table 3.



(a) **(b)**

Fig. 8. The reconstruction results of 7ACT (a) and 7AD1 (b) on noiseless and noisy dataset of SNR=−10 dB.

Table 3. Protein specified hyper-parameters and training/reconstruction time.

Item	4AKE	5A1A	7ACT	7AD1
Protein Mask Radius	0.7	0.5	0.7	0.8
Denoiser Downsample Factor	32	128	32	128
Total Training Time (min)	150	600	150	600

B.3 Reconstruction Results

Results of Remaining Proteins We show that Cryo2SSNet generalizes to different proteins by conducting reconstruction on simulated datasets of 4 proteins. The reconstruction results are summarized in Table 4. For 7ACT and 7AD1, Cryo2SSNet achieves reconstruction resolution of 2.31 Å and 3.38 Å, respectively, on datasets with SNR=−10 dB, which are close to the corresponding Nyquist-Shannon limits. The reconstruction FSC curves are shown in Fig. 8 and visualization results are displayed in Fig. 9.

Results of Different Density Map Resolutions In this section, we show that Cryo2SSNet can generalize to different density map resolutions. This ability is necessary since when handling real datasets, whose underlining density map resolutions are unknown and the reconstruction algorithm should adapt to a width range of it. The protein 4AKE is used as our testbed. Specifically, we

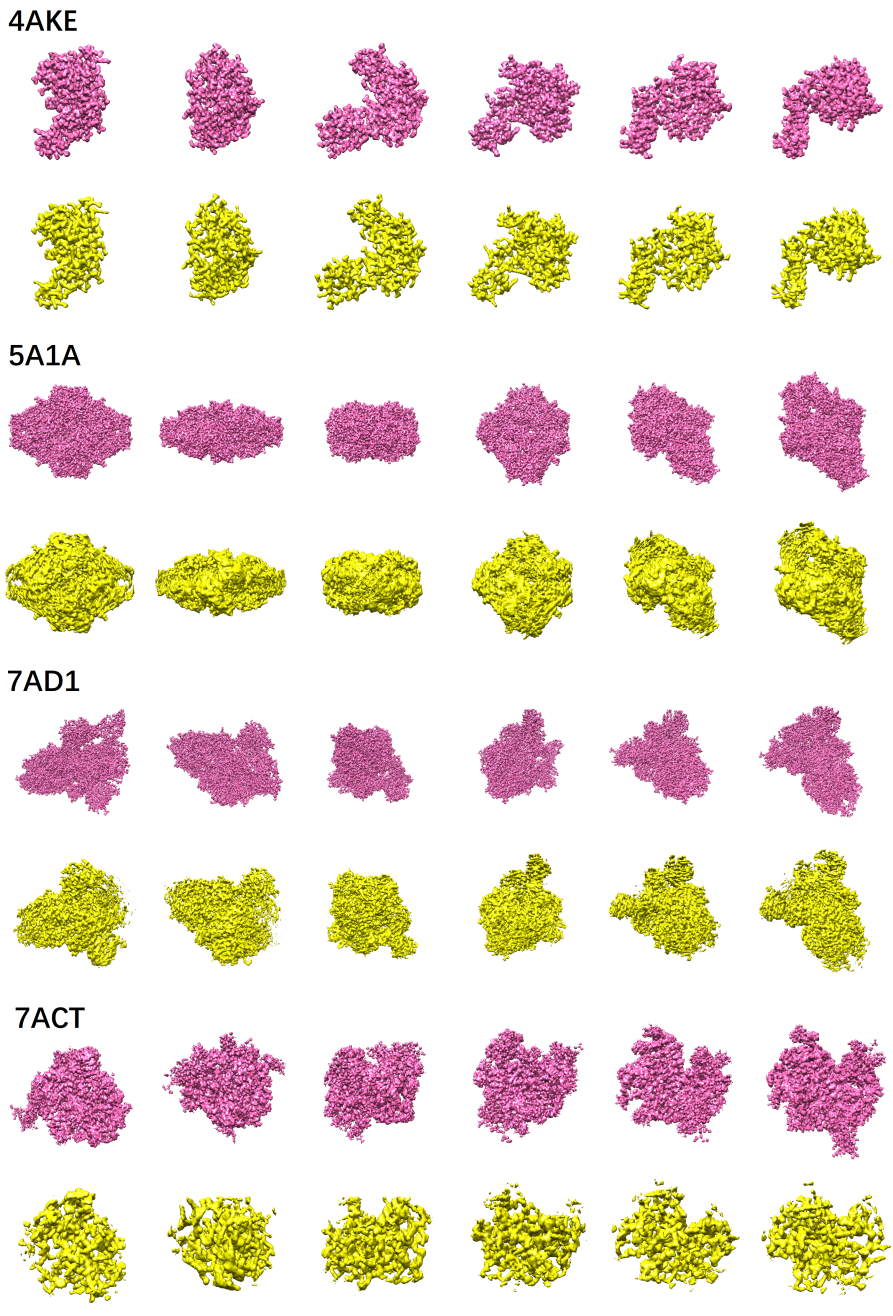
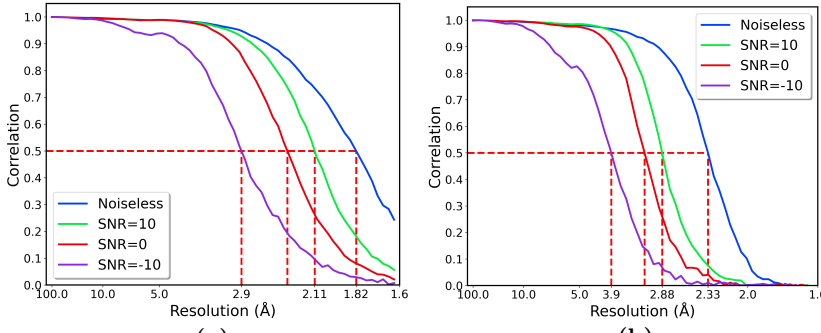


Fig. 9. Visualization of reconstructed density maps. SNR for 5A1A is −20 dB, while those for others are −10 dB. The pink ones are ground truths while the yellow ones are our reconstructions.

Table 4. The reconstruction resolutions of Cryo2SSNet. The results are compared with ones reported by previous proposed methods.

Protein	SNR (dB)	Resolution (Å)	Previous Resolution(Å)
4AKE	∞	1.74	/
	10	1.99	/
	0	1.94	2.57 (CryoPoseNet)
	-10	2.20	/
5A1A	-5.2	2.68	8.30 (CryoGAN)
	-10	3.14	/
	-20	4.64	15.30 (CryoGAN)
7AD1	∞	2.75	/
	-10	3.38	/
7ACT	∞	1.56	/
	-10	2.31	/

conduct reconstruction with the same hyper-parameters described in Section B.2 except creating couple of new datasets with density map resolutions 3.0 Å and 5.0 Å. As shown in Fig. 10, Cryo2SSNet works well among all these settings. It achieves 2.90 Å and 3.90 Å resolution for datasets of SNR=−10 dB with density map resolution 3.0 Å and 5.0 Å, respectively, which degrades an acceptable fraction under gradually worse datasets.



(a) **(b)**

Fig. 10. Reconstruction of 4AKE datasets from density map resolution of 3.0 Å (a) and 5.0 Å (b). Cryo2SSNet performs consistently well and generalizes among different SNR.

Visualization of Loss Landscape To illustrate the reason why it is feasible to perform gradient descent enhancement after beam search rather than from scratch, we visualize the loss landscape when fixing the density map and changing the projection orientation. We pick one image from the 5A1A dataset with SNR=−20 dB and fetch the ground truth pose and density map. While the

quaternion representation of the pose is on a $\mathbb{R}^4$ sphere, two dimensions are perturbed in certain ranges (and normalized) to form a grid. At each grid point, the unbiased ℓ_2 loss between the projection and the original noisy image is calculated, by which a loss landscape is plotted. We change the perturbation range from 2.5 down to 2.5×10^{-3}, and the results are shown in Fig. 11.

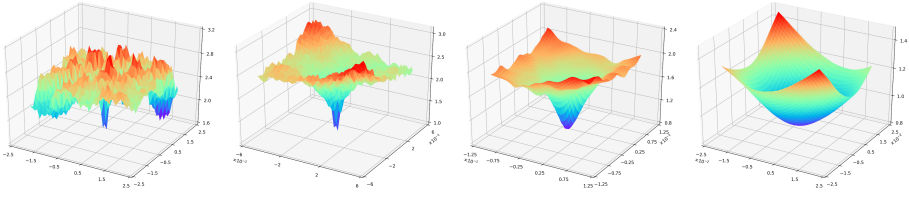


Fig. 11. Visualization of loss landscape. The perturbation ranges are from 2.5 (left most) to 2.5×10^{-3} (right most).

As we can see, the ground truth poses almost locates at the global minimum, while the landscape outside this valley is in a mess. Even the losses at points near the ground truth can be pretty large. When zooming in, we can see the landscape is becoming smooth and even convex, which is a friendly condition for gradient descent. Therefore, DHBS performs gradient descent enhancement after searching into a narrow region, otherwise, it cannot reach a good enough destination.