# CryoSD: Score Distillation for Heterogeneous Cryo-EM Reconstruction

**Anonymous authors**
Paper under double-blind review

## Abstract

Heterogeneous *ab initio* reconstruction in single-particle cryo-electron microscopy (cryo-EM) aims to recover an ensemble of 3D molecular structures from extremely noisy 2D projection images. The task is further complicated by unknown particle orientations and contrast transfer function (CTF) distortions. Existing approaches typically combine variational autoencoders with explicit pose estimation, which becomes unreliable in the presence of severe noise, pronounced conformational variability, and structural symmetries. We introduce CryoSD, a framework for heterogeneous cryo-EM reconstruction that combines ideas from score-based diffusion models, Noisier2Noise, and score distillation sampling (SDS). CryoSD learns a score-based diffusion model over particle images and leverages the learned image distribution for unsupervised denoising, CTF correction, and SDS-based 2D-to-3D reconstruction, eliminating the need for explicit pose estimation. We evaluate CryoSD on four challenging datasets exhibiting continuous and compositional heterogeneity, including IgG-RL and IgG-1D from CryoBench and data from EMPIAR-10295. We compare CryoSD with the state of the art in heterogeneous *ab initio* reconstruction. Our results demonstrate the feasibility of SDS-based heterogeneous cryo-EM reconstruction and suggest CryoSD as a powerful alternative to existing approaches. In particular, CryoSD shows superior performance on sparse heterogeneous data.

## 1 Introduction

Single-particle cryo-electron microscopy (cryo-EM) reconstructs biomolecular structures from large sets of extremely noisy 2D projection images distorted by a contrast transfer function (CTF) (Frank, 2006; Nogales & Scheres, 2015). In a typical cryo-EM experiment, large ensembles of biomolecules are rapidly frozen and imaged in unknown and randomly distributed orientations, yielding projection images of individual particles. Compared to standard tomography, reconstruction is fundamentally more challenging because both the underlying 3D structure and the associated viewing directions must be inferred jointly from highly noisy observations. In the *ab initio* setting, only the projection images and their CTF parameters are available; their relative orientations are unobserved, and no prior information about the underlying volume is assumed. A further difficulty is that many biomolecules are inherently dynamic. Standard reconstruction methods assume that all projection images arise from a single static 3D object, but this assumption is often violated in practice. Instead, different images may correspond to different conformations or compositional states of the macromolecular complex. The goal of heterogeneous *ab initio* reconstruction is therefore to recover the underlying family of 3D densities that generated the observed 2D projections, without assuming that all images correspond to an identical volume.

Most state-of-the-art methods follow the classical cryo-EM paradigm: they alternate between estimating particle poses with respect to a current 3D model and updating the model using the oriented 2D images. In the heterogeneous setting, this problem becomes even more difficult because one must infer not only the pose of each particle but also assign it to a conformational state. Recent methods such as CryoDRGN (Zhong et al., 2020; 2021a), CryoDRGN2 (Zhong et al., 2021b), CryoFIRE (Levy et al., 2022), 3DFlex (Punjani & Fleet, 2023), Opus-DSD (Luo et al., 2023), and CryoDRGN-AI (Levy et al., 2025) address this challenge using variational autoencoders (VAEs) (Kingma & Welling, 2013).



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

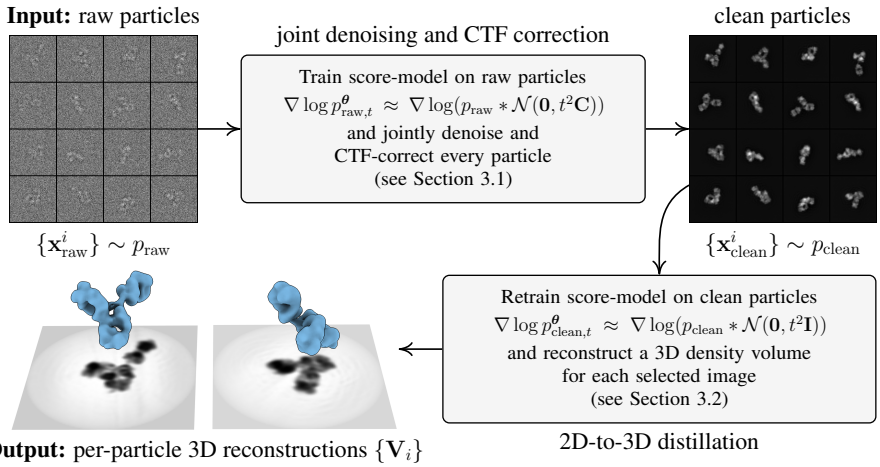



Figure 1: Overview of CryoSD. We train a score-model directly on raw, noise- and CTF-affected particles and use it for joint denoising and CTF correction. We then retrain the model on the restored particles and use the learned clean-image distribution for anchor-conditioned 3D reconstruction. The matched particle grids and reconstructed volumes are actual representative results on IgG-1D.

We approach this problem in a fundamentally different way. Modern generative formulations, such as score-based diffusion models (Song et al., 2020) and related approaches (Ho et al., 2020; Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022), have shown a strong ability to learn complex data distributions from finite sets of samples. While such models are often used for synthesis, our goal is different: we use the learned score not to generate new images, but as a representation of the particle image distribution itself. This perspective enables downstream inference procedures such as score distillation sampling (SDS) (Poole et al., 2022). In this paper, we propose a fully score-based approach for heterogeneous *ab initio* reconstruction in cryo-EM. Existing reconstruction methods typically explain each observation by finding a pose under which a reconstructed volume is consistent with the given image. In contrast, our formulation seeks volumes whose induced distribution of projections agrees with the observed image distribution across orientations. This leads to a fundamentally different reconstruction strategy that avoids explicit pose-search and is in the spirit of CryoGAN-style distribution matching approaches to cryo-EM (Gupta et al., 2021; 2020) based on generative adversarial networks (Goodfellow et al., 2014).

Our method proceeds in two stages, as illustrated in Figure 1. In the first stage, we train a score model directly on the raw, noisy, and CTF-affected particle images $\mathbf{x}_{\text{raw}}$. We then combine ideas from Noisier2Noise (Moran et al., 2020) with an SDS-like optimization that jointly denoises every particle and approximately inverts its CTF. The measured particle remains a fixed anchor under its own CTF, while score-based targets are generated after applying randomly sampled CTFs to the image under optimization. The resulting denoised and CTF-corrected particle images $\mathbf{x}_{\text{clean}}$ are used to further train the score model so that it captures the clean projection-image distribution. In the second stage, we reconstruct a 3D volume $\mathbf{V}_i$ for each chosen clean projection image $\mathbf{x}^i_{\text{clean}}$ by treating $\mathbf{x}^i_{\text{clean}}$ as a 2D anchor and lifting it along the new axis into 3D via anchor-conditioned SDS. This results in a fully unsupervised heterogeneous reconstruction pipeline that relies exclusively on 2D score-based models.

## 2 Background

### 2.1 Heterogeneous *ab initio* reconstruction

Recent neural methods for cryo-EM heterogeneity use latent-variable, VAE-style (Kingma & Welling, 2013) generative models to represent continuous structural variability. Here, the encoder maps 2D particle images to latent conformational states, while the decoder maps each latent to a corresponding 3D structure. The methods differ mainly in the amount of required input. CryoDRGN (Zhong et al., 2020; 2021a), 3DFlex (Punjani & Fleet, 2023), and Opus-DSD (Luo et al., 2023) operate

in a refinement setting: they model conformational variability based on particle images and CTF parameters together with poses obtained from a prior consensus reconstruction run. Thus, while they are powerful tools for heterogeneity analysis, they do not operate fully *ab initio*. In contrast, CryoDRGN2 (Zhong et al., 2021b), CryoFIRE (Levy et al., 2022), and CryoDRGN-AI (Levy et al., 2025) aim to solve the harder fully *ab initio* problem, where particle poses are not provided and must be inferred jointly with structure and heterogeneity directly from the particle stack and CTF parameters. CryoDRGN2 extends the CryoDRGN framework to this setting by jointly searching over poses and conformations. CryoFIRE improves scalability through amortized inference of both pose and conformation. CryoDRGN-AI is the newest release in the CryoDRGN suite, integrating concepts from earlier approaches and tailored for difficult heterogeneous *ab initio* reconstruction.

Established cryo-EM packages like cryoSPARC (Punjani et al., 2017) and RELION (Scheres, 2012) also provide *ab initio* workflows for discrete/compositional heterogeneity typically called multi-class *ab initio* reconstruction. Here, the user needs to select the number of distinct states prior to reconstruction. These workflows are commonly used for 3D particle cleaning or for separating a small number of clearly distinct species or compositional states. Because they represent heterogeneity with a fixed, user-selected number of volumes, they are not designed to directly recover a continuous distribution of structures.

### 2.2 Score-based diffusion models

Score-based diffusion models (SBDMs) (Song et al., 2020) learn a generative model from samples $\{\mathbf{x}^i_{\text{data}}\} \sim p_{\text{data}}$ by estimating the score of Gaussian-mollified versions of the data distribution. Throughout the paper we will follow the SBDM-framework of Karras et al. (2022). For noise level $t > 0$, the mollified density is defined as the convolution $p_t = \mathcal{N}(\mathbf{0}, t^2\mathbf{I}) * p_{\text{data}}$, which approaches the data distribution as $t \to 0$ and becomes approximately Gaussian for sufficiently large $t$. Following Karras et al. (2022), we parameterize the score through a denoiser $D_{\boldsymbol{\theta}}$ as

$$\nabla_{\mathbf{x}} \log p^{\boldsymbol{\theta}}_t(\mathbf{x}) = t^{-2}[D_{\boldsymbol{\theta}}(\mathbf{x}, t) - \mathbf{x}]. \tag{1}$$

The denoiser is trained with denoising score matching (Vincent, 2011; Song et al., 2020; Karras et al., 2022):

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t,\, \mathbf{x}^i_{\text{data}},\, \boldsymbol{\epsilon}} \left[ \lambda(t) \left\| D_{\boldsymbol{\theta}}(\mathbf{x}^i_{\text{data}} + t\boldsymbol{\epsilon}, t) - \mathbf{x}^i_{\text{data}} \right\|_2^2 \right], \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

For a zero-centered dataset with standard deviation $0.5$, the Karras et al. (2022) formulation samples $t$ from the log-normal distribution $\log t \sim \mathcal{N}(-1.2, 1.2^2)$, truncated to $t_{\min} = 0.002$ and $t_{\max} = 80$ (Karras et al., 2022). After training, new samples can be generated by drawing $\mathbf{x}_{t_{\max}} \sim \mathcal{N}(\mathbf{0}, t^2_{\max}\mathbf{I})$ and integrating the probability-flow ODE $\mathrm{d}\mathbf{x}/\mathrm{d}t = -t\nabla_{\mathbf{x}} \log p^{\boldsymbol{\theta}}_t(\mathbf{x})$ backward from $t_{\max}$ to $t_{\min}$. This reverse dynamics progressively removes noise and transports the initial Gaussian sample toward the learned data distribution. While the corresponding reverse dynamics can be used for image generation, our objective is not to synthesize new data. Rather, we use the learned score model as a representation of the particle-image distribution, enabling downstream applications such as denoising, CTF-correction and 3D reconstruction.

#### 2.2.1 Score distillation sampling

Score-based diffusion models trained on text-image pairs have been employed to generate 3D assets without corresponding 3D training data. Building on a text-to-image diffusion model trained on large-scale internet data, Poole et al. (2022) proposed score distillation sampling (SDS) for optimizing a neural radiance field (NeRF) (Mildenhall et al., 2021) to represent a 3D asset that matches a text prompt $\mathbf{y}$. The NeRF parameters $\boldsymbol{\phi}$ are optimized by minimizing a reverse Kullback–Leibler divergence between the mollified distribution of images produced by a differentiable renderer $\mathbf{f}$ and the text-conditioned diffusion model:

$$\min_{\boldsymbol{\phi}} \mathbb{E}_t \left[ t^2 D_{\text{KL}}\big(\mathcal{N}(\mathbf{f}(\boldsymbol{\phi}), t^2\mathbf{I}) \,\|\, p^{\boldsymbol{\theta}}_t(\cdot \mid \mathbf{y})\big) \right], \qquad t \sim \mathcal{U}[a, b]. \tag{3}$$

Here, $\mathbf{f}$ renders the 3D representation from a randomly sampled orientation.[1] Consequently, the learned 2D diffusion model provides the optimization signal for a 3D representation even though

[1] We write the SDS objective using the notation and noise convention of this paper and set its additional time-dependent weighting to one.

it was not trained on 3D data. Although Poole et al. (2022) generate renderable surfaces from text prompts rather than volumetric molecular densities, this absence of 3D training data motivates our adaptation of SDS to heterogeneous cryo-EM. We use SDS in the two steps of our pipeline. First, we distill CTF-free particle images $\mathbf{x}_{\mathrm{clean}}$ from a model trained on noisy CTF-affected images $\mathbf{x}_{\mathrm{raw}}$. Afterwards, we distill voxelized 3D density maps from a score model trained on clean 2D particle images $\mathbf{x}_{\mathrm{clean}}$. The former operation is described in Section 3.1, and the latter in Section 3.2. For additional details on related work in unsupervised denoising, see Section A in the Appendix.

# 3 METHODS

We propose a full *ab initio* algorithm for heterogeneous 3D reconstruction based on a stack of raw particle images $\mathbf{x}_{\mathrm{raw}}$ with corresponding CTF parameters, as illustrated in Figure 1. The stack $\mathbf{x}_{\mathrm{raw}}$ contains both measurement noise and CTF effects, while $\mathbf{x}_{\mathrm{clean}}$ denotes the corresponding stack of noise and CTF-free projection images. For each particle $i$, the cryo-EM forward model is

$$\mathbf{x}^i_{\mathrm{raw}} = \mathcal{F}^{-1}\big\{\mathrm{CTF}_i\mathcal{F}\big\{\mathbf{x}^i_{\mathrm{clean}}\big\}\big\} + \sigma\boldsymbol{\epsilon}_i, \quad \mathbf{x}^i_{\mathrm{clean}} = (\mathcal{T}_i \circ \Pi \circ \mathcal{R}_i)(\mathbf{V}_i), \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{C}). \tag{4}$$

Here, $\mathcal{R}_i$ rotates the associated volume $\mathbf{V}_i$ in 3D, $\Pi$ projects it into 2D, and $\mathcal{T}_i$ applies an in-plane translation. We write $\mathbf{x}^i_{\mathrm{ctf}} = \mathcal{A}(\mathbf{x}^i_{\mathrm{clean}}, \mathrm{CTF}_i) = \mathcal{F}^{-1}\{\mathrm{CTF}_i\mathcal{F}\{\mathbf{x}^i_{\mathrm{clean}}\}\}$ for the noise-free but CTF-affected image. Here, covariance $\mathbf{C}$ is normalized, while $\sigma$ controls the noise amplitude. The typical cryo-EM forward model assumes white noise $\mathbf{C} = \mathbf{I}$. Then the noise in each pixel is independent of other pixels, but identically Gaussian-distributed. However, some experimental data contain colored noise, where pixels are locally correlated. For every $\mathbf{x}^i_{\mathrm{raw}}$, we first approximately restore its corresponding $\mathbf{x}^i_{\mathrm{clean}}$ by jointly denoising the raw particle and inverting its CTF before 2D-to-3D reconstruction.

## 3.1 DENOISING AND CTF CORRECTION

Let $p_{\mathrm{raw}}$ be the distribution of the raw particle images and $p_{\mathrm{raw},t} = \mathcal{N}(\mathbf{0}, t^2\mathbf{C}) * p_{\mathrm{raw}}$ its $t$-mollified distributions. During **training**, we form $\mathbf{z}^i_t = \mathbf{x}^i_{\mathrm{raw}} + t\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, and train $D_{\boldsymbol{\theta}}$ using the Karras-style denoising score-matching objective. The denoiser parametrizes the learned score as $\nabla_{\mathbf{z}^i_t} \log p^{\boldsymbol{\theta}}_{\mathrm{raw},t}(\mathbf{z}^i_t) = (t^2\mathbf{C})^{-1}[D_{\boldsymbol{\theta}}(\mathbf{z}^i_t, t) - \mathbf{z}^i_t]$. All raw particle images are standardized across the complete stack to zero mean and unit variance. Effective DSM training requires substantial augmentation, since the distribution of particle images should ideally be invariant under in-plane rotations, reflections and small translations. To avoid interpolation artifacts in the raw noisy and CTF-affected images, we use only uniformly sampled multiples of $90^\circ$ together with random reflections. The denoiser uses the UNet2DModel architecture from the diffusers library of von Platen et al., whose convolutional structure is approximately translation equivariant. Training otherwise follows Karras et al. (2022), but we emphasize the low-noise regime relevant to denoising rather than image synthesis. Specifically, for $u \sim \mathcal{N}(0, 1)$ we sample $t = \mathrm{clip}_{[t_{\min}, t_{\max}]}[1.05t_{\max}\,\mathrm{sigmoid}(u) - 0.025t_{\max}]$, with $t_{\min} = 0.002$ and $t_{\max} = 2$ for the raw-particle DSM training.

For the datasets capsidHA and EMPIAR-10295 (see Section 4) containing colored noise we manually picked small particle-free regions (close to the edge of the images). We mean-center every crop, average their individually normalized periodograms, and radialize the result to estimate the spatial autocorrelation encoded by $\mathbf{C}$. The scale $\sigma$ was set manually by visual inspection. For data with white noise we simply set $\mathbf{C} = \mathbf{I}$.

During **inference** we want to use the score model trained on $\mathbf{x}_{\mathrm{raw}}$ to obtain $\mathbf{x}_{\mathrm{clean}}$ in a fully unsupervised way. Let $p_{\mathrm{ctf}}$ denote the distribution of particles that are noise-free but affected by the CTF. Under our observation model, $p_{\mathrm{raw},0} = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{C}) * p_{\mathrm{ctf}}$. Evaluating the raw model at $t = \alpha\sigma$ with $\alpha > 0$ therefore induces the model $p^{\boldsymbol{\theta}}_{\mathrm{ctf},\tau} \equiv p^{\boldsymbol{\theta}}_{\mathrm{raw},\alpha\sigma}$ of the $\tau$-mollified CTF-image distribution, where $\tau^2 = (1 + \alpha^2)\sigma^2$. For a CTF row $j$ sampled uniformly from the dataset, we can perturb $\mathcal{A}(\mathbf{x}^i_{\mathrm{clean}}, \mathrm{CTF}_j)$ by independent noise components with scales $\sigma$ and $\alpha\sigma$. Their covariances add, hence $\mathbf{z}^{i,j}_{\alpha\sigma} \sim \mathcal{N}(\mathcal{A}(\mathbf{x}^i_{\mathrm{clean}}, \mathrm{CTF}_j), \tau^2\mathbf{C})$, while the denoiser is evaluated as $D_{\boldsymbol{\theta}}(\mathbf{z}^{i,j}_{\alpha\sigma}, \alpha\sigma)$.

For the particle-specific anchor, we draw $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, set $\mathbf{z}^i_{\alpha\sigma} = \mathbf{x}^i_{\mathrm{raw}} + \alpha\sigma\boldsymbol{\epsilon}$, and use the score-based diffusion estimate $\mathbf{x}^i_{\mathrm{ctf}} \approx \mathbf{z}^i_{\alpha\sigma} + \tau^2\mathbf{C}\nabla_{\mathbf{z}^i_{\alpha\sigma}} \log p^{\boldsymbol{\theta}}_{\mathrm{ctf},\tau}(\mathbf{z}^i_{\alpha\sigma})$. This is the score-based form of the Noisier2Noise estimator and yields a noise-free estimate that remains distorted by $\mathrm{CTF}_i$. Defining

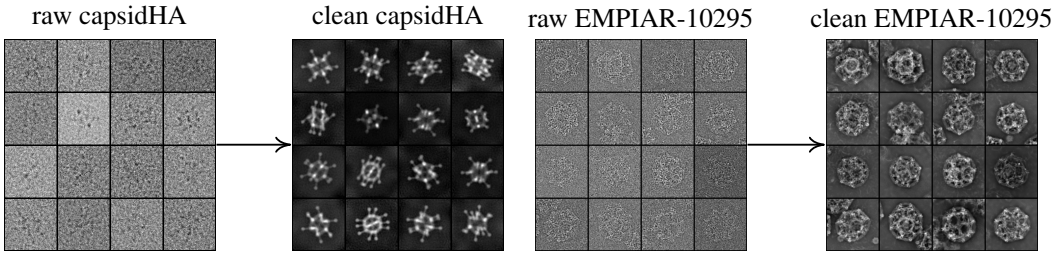



Figure 2: Randomly drawn examples of joint denoising and CTF correction on capsidHA and EMPIAR-10295. Each clean particle corresponds to the raw particle at the same grid position.

$q_\tau^{i,j} = \mathcal{N}(\mathcal{A}(\mathbf{x}^i_{\text{clean}}, \text{CTF}_j), \tau^2\mathbf{C})$, we minimize

$$\mathcal{L}^i_{\text{2D}}(\mathbf{x}^i_{\text{clean}}) = w_s\tau^2\,\mathbb{E}_j\left[D_{\text{KL}}\big(q_\tau^{i,j} \,\|\, p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\big)\right] + \frac{w_a}{2}\left\|\mathcal{A}(\mathbf{x}^i_{\text{clean}}, \text{CTF}_i) - \mathbf{x}^i_{\text{ctf}}\right\|_2^2. \tag{5}$$

The reverse-KL term makes the candidate plausible under the learned image distribution, whereas the anchor uses the particular $\text{CTF}_i$ associated with $\mathbf{x}^i_{\text{raw}}$ and ties the solution to that particle. Writing $\mathbf{J}^{i,j}_{\mathcal{A}} = \partial\mathcal{A}(\mathbf{x}^i_{\text{clean}}, \text{CTF}_j)/\partial\mathbf{x}^i_{\text{clean}}$, the gradient is

$$\begin{aligned}\nabla_{\mathbf{x}^i_{\text{clean}}}\mathcal{L}^i_{\text{2D}} = &-w_s\tau^2\,\mathbb{E}_{j,\mathbf{z}^{i,j}_{\alpha\sigma}}\left[(\mathbf{J}^{i,j}_{\mathcal{A}})^\top\nabla_{\mathbf{z}^{i,j}_{\alpha\sigma}}\log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}(\mathbf{z}^{i,j}_{\alpha\sigma})\right]\\ &+ w_a(\mathbf{J}^{i,i}_{\mathcal{A}})^\top\left[\mathcal{A}(\mathbf{x}^i_{\text{clean}}, \text{CTF}_i) - \mathbf{x}^i_{\text{ctf}}\right].\end{aligned} \tag{6}$$

Throughout optimization, the model parameters $\boldsymbol{\theta}$ remain frozen. We follow typical stochastic gradient descent (SGD) and approximate the expectation with $M$ independently sampled CTFs from the dataset and update $\mathbf{x}^i_{\text{clean},k+1} = \mathbf{x}^i_{\text{clean},k} - \beta\nabla_{\mathbf{x}^i_{\text{clean},k}}\mathcal{L}^i_{\text{2D}}$. We initialize SGD by an approximate inversion of $\text{CTF}_i$ with a Wiener filter to obtain the initial clean image $\mathbf{x}^i_{\text{clean},0} = \mathcal{F}^{-1}\{\text{CTF}_i(\text{CTF}_i^2 + \gamma)^{-1}\mathcal{F}\{\mathbf{x}^i_{\text{ctf}}\}\}$ (we always used $\gamma = 0.001$). Detailed derivation of the gradient is provided in Appendix C.1. After $S$ steps, $\mathbf{x}^i_{\text{clean},S} \approx \mathbf{x}^i_{\text{clean}}$. Repeating the procedure for every raw particle yields the restored stack $\mathbf{x}_{\text{clean}}$. Throughout the experiments we used a fixed learning rate of $\beta = 0.5$, equal weighting of the score and anchor contributions $w_s = w_a = 1$, batch size of $M = 1$ and noise augmentation strength $\alpha = 0.25$. For more experimental details see Table 1 in the Appendix. Representative examples of our denoising and CTF-correction procedure can be found in Figures 1 and 2 .

### 3.2 3D RECONSTRUCTION

After processing the raw particles $\mathbf{x}_{\text{raw}}$, we retrain the same diffusion-model architecture on the denoised and CTF-corrected stack $\mathbf{x}_{\text{clean}}$. Before training, the pixel values are standardized across the complete restored stack to zero mean and unit variance. For the clean-image model, we augment with arbitrary in-plane rotations and reflections. After rotation, pixels outside the inscribed circular mask are set to the normalized mean background value to reduce boundary artifacts. Once the 2D diffusion model has been trained on $\mathbf{x}_{\text{clean}}$, it represents the $t$-mollified clean-image distribution $p^{\boldsymbol{\theta}}_{\text{clean},t}$, and we use it to infer the 3D volume associated with each selected image $\mathbf{x}^i_{\text{clean}}$. We denote the voxelized 3D volume corresponding to the pair $(\mathbf{x}^i_{\text{raw}}, \mathbf{x}^i_{\text{clean}})$ by $\mathbf{V}_i$. Given a rotation $\mathbf{R}$ sampled from a reconstruction distribution $\rho$, the volume defines a smoothed distribution over possible 2D projections,

$$q_t(\cdot \mid \mathbf{V}_i, \mathbf{R}) := \mathcal{N}(\mathbf{f}(\mathbf{V}_i, \mathbf{R}), t^2\mathbf{I}), \tag{7}$$

where $\mathbf{f}(\mathbf{V}_i, \mathbf{R})$ rotates $\mathbf{V}_i$ and projects it onto the image plane. For reconstruction, we optimize $\mathbf{V}_i$ by minimizing the expected reverse KL divergence between the smoothed projection distribution induced by the volume and the learned mollified clean-image distribution $p^{\boldsymbol{\theta}}_{\text{clean},t}$. However, heterogeneous datasets contain projections of multiple conformational states. We therefore select the restored particle $\mathbf{x}^i_{\text{clean}}$ as an anchor, fix its orientation to the identity, and minimize

$$\mathcal{L}^i_{\text{3D}}(\mathbf{V}_i) = w_s t^2\mathbb{E}_{\mathbf{R}\sim\rho}\left[D_{\text{KL}}\big(q_t(\cdot \mid \mathbf{V}_i, \mathbf{R}) \,\|\, p^{\boldsymbol{\theta}}_{\text{clean},t}\big)\right] + \frac{w_a}{2}\left\|\mathbf{f}(\mathbf{V}_i, \mathbf{I}) - \mathbf{x}^i_{\text{clean}}\right\|_2^2. \tag{8}$$

The first term drives random volume projections toward modes of the learned clean-image distribution. The second term selects the conformation depicted by the anchor and fixes the otherwise arbitrary global rotational gauge. Writing $\mathbf{J}(\mathbf{V}_i, \mathbf{R}) = \partial \mathbf{f}(\mathbf{V}_i, \mathbf{R})/\partial \mathbf{V}_i$, the voxel update follows the gradient

$$\begin{aligned}\nabla_{\mathbf{V}_i} \mathcal{L}^i_{\text{3D}}(\mathbf{V}_i) = w_s \mathbb{E}_{\mathbf{R}\sim\rho,\boldsymbol{\epsilon}} \left[ \mathbf{J}(\mathbf{V}_i, \mathbf{R})^\top \left( \mathbf{f}(\mathbf{V}_i, \mathbf{R}) - D_{\boldsymbol{\theta}}(\mathbf{f}(\mathbf{V}_i, \mathbf{R}) + t\boldsymbol{\epsilon}, t) \right) \right] \\ + w_a \mathbf{J}(\mathbf{V}_i, \mathbf{I})^\top \left( \mathbf{f}(\mathbf{V}_i, \mathbf{I}) - \mathbf{x}^i_{\text{clean}} \right).\end{aligned} \tag{9}$$

Throughout reconstruction, the denoiser parameters $\boldsymbol{\theta}$ remain fixed. We sample a small batch of rotations at every step, backproject the score and anchor residuals and update the voxels (see the respective algorithm in the Appendix Section B). Because the reverse KL divergence is mode seeking, a single volume is encouraged to match particular modes of the learned projection distribution instead of explaining all particle images at once. This property is well suited to heterogeneous reconstruction, where one conformation contributes only a subset of all observed projections. To make it possible to recover every conformation represented in the dataset, we reconstruct a volume $\mathbf{V}_i$ for each selected anchor $\mathbf{x}^i_{\text{clean}}$, which in turn corresponds to the original raw particle $\mathbf{x}^i_{\text{raw}}$. The complete derivation of Equation 9 is given in Appendix C.2. For the reconstruction algorithm the particle inside $\mathbf{x}^i_{\text{clean}}$ does not need to be centered. An off-center particle will result in an off-center volume, which is not a problem, as long as the full particle is covered by the anchor image. During reconstruction we do not sample translations as they do not provide further signal about the volume during optimization. The rotation distribution $\rho$ need not be uniform on $\mathrm{SO}(3)$. For both IgG datasets (see Section 4), we draw uniformly from $\mathrm{SO}(3)$ throughout optimization. For the more difficult capsidHA and EMPIAR-10295 datasets, we obtained more stable reconstructions with a restricted progressive distribution. Specifically, we first draw the quaternion $\mathbf{q}$ of a uniform random rotation and contract it toward the identity as $\mathbf{q}_p = \mathrm{Slerp}(\mathbf{q}_{\text{id}}, \mathbf{q}, p)$. At $p = 0$, all samples equal the identity. For $0 < p < 1$, the rotations are contracted toward the identity and at $p = 1$, the original uniform distribution on $\mathrm{SO}(3)$ is recovered. We increase $p$ linearly from $0.3$ to $0.6$, so the maximum rotation angle grows from $0.3\pi = 72^\circ$ to $0.6\pi = 108^\circ$ during reconstruction.

## 4 Results

We show that our approach is more than just a conceptual proposal by benchmarking CryoSD across a broad set of test scenarios, including compositional and conformational heterogeneity, particles with signal-to-noise ratios (SNR) as low as $0.005$, both white and colored noise, and experiments on simulated as well as real data. For each test dataset, we train a separate UNet model. Across all models, the UNet architecture consists of three downsampling and three upsampling blocks with channel sizes 64, 128, and 256, respectively, and uses attention with eight heads in the lowest-resolution blocks, implemented with the diffusers library of von Platen et al. We primarily compare against CryoDRGN-AI (v4.3, Aug. 2026) which is the most recent and strongest directly comparable baseline for fully *ab initio* heterogeneous reconstruction (Levy et al., 2025; Jeon et al., 2024). Following the procedure described in CryoBench (Jeon et al., 2024), we ran CryoDRGN-AI with its default configuration, modifying only the number of training epochs from the default 30 to 100 to allow sufficient time for convergence. For synthetic data, we use per-image Fourier shell correlation (FSC), as in CryoBench (Jeon et al., 2024). In per-image FSC we draw 100 random particles from the input stack $\mathbf{x}_{\text{raw}}$. For each selected particle, we reconstruct the conformation depicted in that image, align it with the corresponding ground-truth volume using UCSF ChimeraX (Meng et al., 2023), and compute the FSC (without masking). Per-image FSC is the most comprehensive way to measure the reconstruction performance in the heterogeneous setting. However, per-image FSC can only be evaluated for synthetic data as the correspondence between each image and its ground-truth conformation must be known. We tested and compared our approach on four challenging datasets: the IgG-RL and IgG-1D datasets from CryoBench (Jeon et al., 2024), the capsidHA dataset and the EMPIAR-10295 dataset.

The **IgG-RL** and **IgG-1D** benchmark proposed by Jeon et al. (2024) each contains 100k particle images of the human immunoglobulin G (IgG) antibody complex at $128 \times 128$ resolution and 3 Å/pixel, capturing different shapes of its flexible linker. While IgG-1D conformations represent a simple one-dimensional $360^\circ$ motion generated by uniformly rotating a single Fab-domain hinge dihedral angle, IgG-RL models more complex disordered motion by sampling the backbone dihedral angles of a five-residue linker from Ramachandran distributions, producing randomly oriented Fab conformations (Jeon et al., 2024). Here each of the 100 conformations per dataset contributes exactly

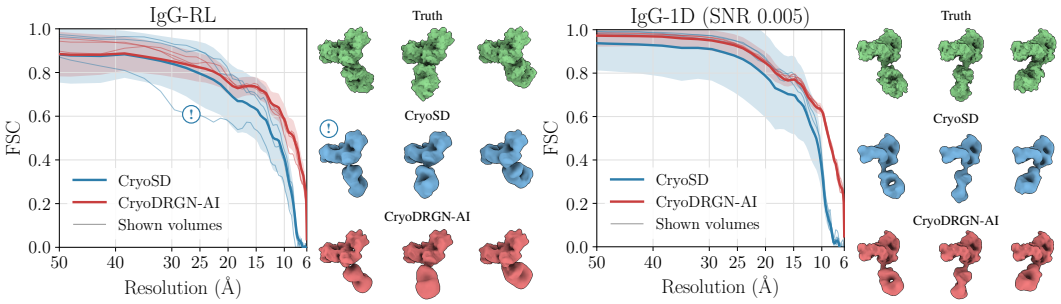



Figure 3: Average FSC curves (from a total of 100) and representative reconstructions for IgG-RL and IgG-1D. The left pair shows IgG-RL, while the right pair shows IgG-1D. Shaded regions show one standard deviation over 100 reconstructions, while thin curves correspond to the displayed volumes. The circled exclamation marks identify the same representative failure case in the FSC plot and reconstruction montage.

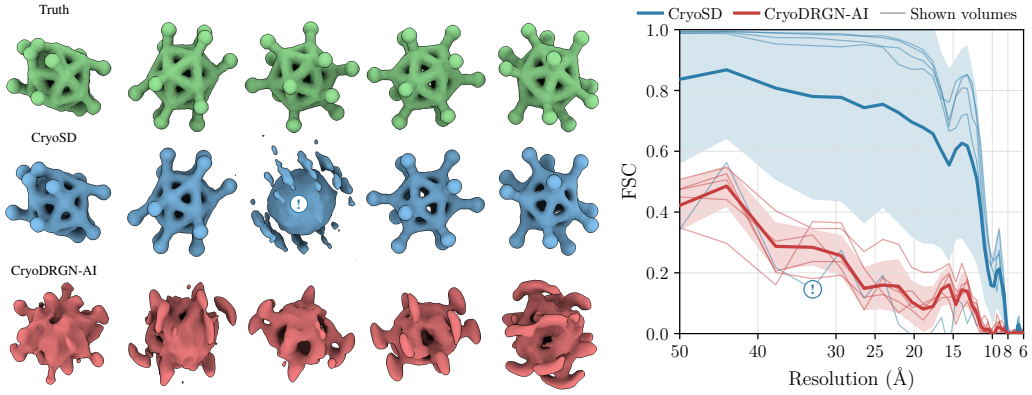



Figure 4: Reconstruction results on capsidHA. The left panel shows representative ground-truth volumes and reconstructions from CryoSD and CryoDRGN-AI. The right panel shows the mean FSC curves (from a total of 100), one-standard-deviation bands, and the curves corresponding to the displayed volumes. Circled exclamation marks identify the same representative failure case in the montage and FSC plot.

1k particle images to the particle stack. Both datasets contain white noise. We used the IgG-1D version with an SNR of 0.005, while IgG-RL has an SNR of 0.01. Following the approach used in CryoBench, we sampled one random image per conformation for the per-image FSC instead of sampling the images completely at random from the input particle stack. For *ab initio* reconstruction with CryoSD, we used the procedure outlined in Section 3; a detailed specification of all parameter settings is provided in Table 1 in the Appendix. The results of both datasets for our method and CryoDRGN-AI are presented in Figure 3.

The **capsidHA** model consists of an icosahedral virus-like core with twelve heavy external arms whose mirrored tilt defines a continuous opening motion. We designed the model to contain high mass in the flexible regions compared to the static core such that the heterogeneous component contributes strongly to the particle images and reconstruction metric. The dataset contains 25,000 projections of 200 conformations at $104 \times 104$ pixels and 3 Å/pixel. Rather than adding white noise, we use colored noise whose radial power spectral density is estimated from manually selected particle-free background crops from EMPIAR-10454, an experimental dataset with visibly spatially correlated noise. Full simulation and noise-estimation details are given in Appendix D.1. Heterogeneous reconstruction results of CryoSD and CryoDRGN-AI can be seen in Figure 4.

Finally, we evaluate CryoSD on 8183 raw particles from **EMPIAR-10295**, which contains a mixture of five coexisting clathrin-cage architectures: the 28-triskelion mini coat, 32-triskelion sweet potato, 36-triskelion D6 barrel, 36-triskelion tennis ball, and big apple (see Figure 5). The preprocessing

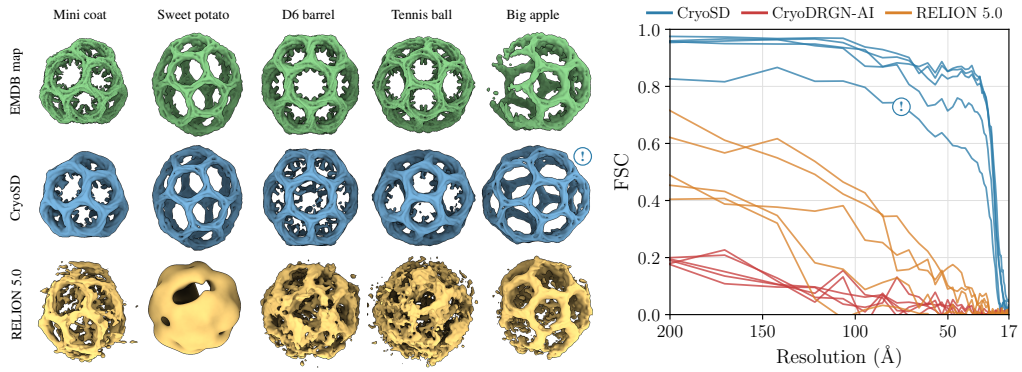



Figure 5: Reconstruction results on EMPIAR-10295. The left panel compares the published EMDB maps with the reconstructions from CryoSD and RELION 5.0. The right panel shows the corresponding unmasked FSC curves for CryoSD, CryoDRGN-AI, and RELION 5.0. The exclamation marks identify the corresponding highlighted case in the reconstruction montage and FSC plot.

is detailed in the Appendix D.2. In the original study by Morris et al. (2019), the authors did not determine the cages *ab initio*. Instead, because of the challenging heterogeneous reconstruction problem they encountered, they carried out supervised 3D classification using ten theoretically predicted, geometrically plausible *in silico* cage models. However, this kind of extensive prior knowledge about the structures being imaged is typically not available. We evaluated whether CryoSD, CryoDRGN-AI, and RELION are capable of reconstructing the mixed cage population *ab initio*. As the deposited EMPIAR-10295 dataset reports a discrete mixture of conformations, we performed multi-class *ab initio* reconstruction using RELION 5.0 (Scheres, 2012) under default settings. Here, we specified the assumed number of five classes and ran optimization for 160 iterations (default is 20) to provide sufficient time for convergence. For RELION we tested all possible $5 \times 5$ alignments between the ground truth volumes and the five RELION reconstructions to find the best assignment. CryoDRGN-AI was unable to reconstruct any sensible cage structure (see Figure 6). The CryoDRGN-AI log files and loss curve appeared normal and did not show any signs that would suggest improper use. To report quantitative FSC values we used the representative central volume reported by CryoDRGN-AI at the last checkpoint and aligned it to the five deposited EMDB maps. For CryoSD we followed the procedure described in Section 3. During 3D reconstruction we randomly picked anchors from the restored particle stack $\mathbf{x}_{\text{clean}}$ and generated one anchor-conditioned volume $\mathbf{V}_i$ per image. We selected reconstructions when they formed a coherent cage and resolve triskelion-like features. Then we matched them to the published maps by their coarse geometry. Figure 5 presents selected outcomes rather than an estimate that every anchor produces a high-quality cage. Indeed, a small fraction of generated volumes collapse towards unstructured spherical densities, and some form fragmented cages (see Figure 6). For each matched volume, we aligned the reconstruction to the corresponding EMDB map using ChimeraX (Meng et al., 2023), then computed the unmasked FSC (see Figure 5). The big apple comparison exposes a limitation of treating the deposited map as ground truth. The theoretical big apple reference used by Morris et al. (2019) contained 38 triskelia, whereas their reconstruction produced a volume containing 37; this difference is stated explicitly in the original study (Morris et al., 2019), and the resulting 37-triskelion density was deposited as EMD-0120. Our selected *ab initio* reconstruction is instead consistent with the complete theoretical 38-triskelion cage (see Section D.2 in the Appendix about the big apple topology verification). Thus, CryoSD reconstructs the complete big apple cage fully *ab initio*. Accordingly, the reduced FSC for big apple (marked in Figure 5) should be interpreted as structural disagreement between a 38-triskelion reconstruction and the incomplete 37-triskelion deposited reference and not as failure.

## 5 Discussion

CryoSD reconstructs the conformational dynamics of both IgG datasets, although CryoDRGN-AI achieves higher FSC at high spatial frequencies and therefore recovers more fine structural detail overall. The spatial distribution of this detail differs in IgG-RL: CryoDRGN-AI reconstructs the rigid core sharply while the flexible region remains blurred, whereas CryoSD produces a more

uniform reconstruction quality across the rigid and flexible parts of the antibody. The relative performance changes on the more difficult capsidHA and EMPIAR-10295 experiments, which combine comparatively sparse heterogeneous sampling, colored experimental or experimentally derived noise, and structures with challenging symmetries or pseudo-symmetries. In these settings, CryoSD clearly outperforms CryoDRGN-AI and the evaluated multi-class *ab initio* baseline from the RELION package. Here the pose-search-based approach clearly fails on the capsidHA and EMPIAR-10295 experiments. We hypothesize that this is due to the joint difficulty of estimating accurate image-to-volume orientations while simultaneously accounting for structural heterogeneity. This problem is particularly severe for highly symmetric structures such as the capsidHA core, where many viewing directions produce similarly plausible projections, leading to a highly multi-modal pose-estimation landscape. Pose-search-based algorithms aim to find a volume such that each observed image can be explained by some pose of that volume. By contrast, our method seeks volumes whose full distribution of projections remains consistent with the learned particle-image distribution across viewing directions.

**Limitations.** Potential advantages of our approach do not eliminate optimization failures: a small number of capsidHA and EMPIAR-10295 reconstructions collapse toward unstructured spherical densities (see Figures 4 and 6). We attribute these cases to instability of the non-convex anchor-conditioned 2D-to-3D optimization, which can become trapped in a poor local optimum. Additionally, the CryoSD failure marked by an exclamation mark in Figure 3 is visually plausible but does not recover the exact conformation depicted by its anchor image. It is representative of reconstructions for which $\mathbf{V}_i$ lies near the learned structural distribution but does not correspond sufficiently closely to the particular $\mathbf{x}^i_{\text{clean}}$ used as its anchor. A disadvantage of CryoSD compared to CryoDRGN-AI is runtime. For the two IgG datasets with comparable results between the two methods, CryoDRGN-AI took $\sim 7$ h for training whereas CryoSD took considerably longer, with $\sim 10.8$ h + 13 h for DSM-training on the raw and clean image distributions, respectively (single A100 GPU with 80 GB of VRAM). Additionally, CryoSD needed $\sim 0.36$ s per image for denoising and CTF correction plus $\sim 1.9$ min per volume reconstruction, whereas a forward pass through the CryoDRGN-AI VAE to generate a volume required negligible additional time. However, we want to emphasize that the inference time of CryoSD can be considerably lowered through parallelization. That said, we do not view this runtime overhead as a major practical barrier, since cryo-EM studies are typically carried out on timescales of weeks to months rather than hours.

## 6 Conclusion

We present CryoSD, a fully score-distillation-based framework for heterogeneous *ab initio* reconstruction in cryo-EM. Heterogeneous *ab initio* reconstruction remains a challenging and largely unresolved task in cryo-EM, particularly for highly flexible macromolecular assemblies under sparse data regimes. Existing heterogeneous reconstruction pipelines typically rely on VAE-driven latent-variable models coupled with conventional pose-search procedures. In contrast, we propose a fundamentally different methodology to tackle this problem. Empirical evaluations suggest that our score-distillation-based approach offers a strong alternative paradigm for heterogeneous cryo-EM reconstruction, with specific emphasis on sparse heterogeneous datasets.

## Reproducibility statement

The main restoration and reconstruction objectives are specified in Equations 5 and 8. The appendix provides the corresponding gradient derivations. Alongside the manuscript we provide the source code and dataset-specific configuration files used for all reported experiments.

## AI use statement

The central research questions, ideas, algorithms, and proofs presented in this work were developed by the authors. Generative AI tools (GPT-5.6 Sol) were used to assist with implementation and debugging. AI tools were also used to draft and edit the manuscript. All AI-assisted text and code were reviewed and verified by the authors. The authors take full responsibility for the complete content of this submission, including all material produced with the assistance of generative AI.

## REFERENCES


Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Tristan Bepler, Kotaro Kelley, Alex J Noble, and Bonnie Berger. Topaz-Denoise: general deep denoising models for cryoEM and cryoET. *Nature communications*, 11(1):5208, 2020.

Joachim Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. Oxford University Press, 2006.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Harshit Gupta, Thong H Phan, Jaejun Yoo, and Michael Unser. Multi-CryoGAN: Reconstruction of continuous conformations in Cryo-EM using generative adversarial networks. In *European Conference on Computer Vision*, pp. 429–444. Springer, 2020.

Harshit Gupta, Michael T McCann, Laurene Donati, and Michael Unser. CryoGAN: A new reconstruction paradigm for single-particle cryo-EM via deep adversarial learning. *IEEE Transactions on Computational Imaging*, 7:759–774, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Minkyu Jeon, Rishwanth Raghu, Miro Astore, Geoffrey Woollard, Ryan Feathers, Alkin Kaz, Sonya M Hanson, Pilar Cossio, and Ellen D Zhong. CryoBench: Diverse and challenging datasets for the heterogeneity problem in cryo-EM. *Advances in Neural Information Processing Systems*, 37:89468–89512, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Kwanyoung Kim and Jong Chul Ye. Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 34:864–874, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2129–2137, 2019.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018.

Axel Levy, Gordon Wetzstein, Julien NP Martel, Frederic Poitevin, and Ellen Zhong. Amortized inference for heterogeneous reconstruction in cryo-em. *Advances in neural information processing systems*, 35:13038–13049, 2022.

Axel Levy, Rishwanth Raghu, J Ryan Feathers, Michal Grzadkowski, Frédéric Poitevin, Jake D Johnston, Francesca Vallese, Oliver Biggs Clarke, Gordon Wetzstein, and Ellen D Zhong. CryoDRGN-AI: neural ab initio reconstruction of challenging cryo-EM and cryo-ET datasets. *Nature Methods*, 22(7):1486–1494, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Zhenwei Luo, Fengyun Ni, Qinghua Wang, and Jianpeng Ma. OPUS-DSD: deep structural disentanglement for cryo-EM single-particle analysis. *Nature Methods*, 20(11):1729–1738, 2023.

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

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12064–12072, 2020.

Kyle L Morris, Joseph R Jones, Mary Halebian, Shenping Wu, Michael Baker, Jean-Paul Armache, Amaurys Avila Ibarra, Richard B Sessions, Alexander D Cameron, Yifan Cheng, et al. Cryo-em of multiple cage architectures reveals a universal mode of clathrin self-assembly. *Nature structural & molecular biology*, 26(10):890–898, 2019.

Eva Nogales and Sjors H. W. Scheres. Cryo-EM: A unique tool for the visualization of macromolecular complexity. *Molecular Cell*, 58(4):677–689, 2015.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Ali Punjani and David J Fleet. 3DFlex: determining structure and motion of flexible proteins from cryo-EM. *Nature Methods*, 20(6):860–870, 2023.

Ali Punjani, John L Rubinstein, David J Fleet, and Marcus A Brubaker. cryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination. *Nature methods*, 14(3):290–296, 2017.

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

Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. *Advances in neural information processing systems*, 31, 2018.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Structura Biotechnology Inc. Job: Micrograph Denoiser (BETA). `https://guide.cryosparc.com/processing-data/all-job-types-in-cryosparc/exposure-curation/job-micrograph-denoiser-beta`, 2026. cryoSPARC Guide. Accessed: 2026-03-03.

Dimitry Tegunov and Patrick Cramer. Real-time cryo-electron microscopy data preprocessing with warp. *Nature methods*, 16(11):1146–1152, 2019.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, Steven Liu, William Berman, Yiyi Xu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. URL `https://github.com/huggingface/diffusers`.

Thorsten Wagner and Stefan Raunser. The evolution of SPHIRE-crYOLO particle picking and its application in automated cryo-EM processing workflows. *Communications biology*, 3(1):61, 2020.

Garrett Wright, Joakim Andén, Vineet Bansal, Junchao Xia, Chris Langfield, Josh Carmichael, Kris Sowattanangkul, Robbie Brook, Yunpeng Shi, Ayelet Heimowitz, Gabi Pragier, Itay Sason, Amit Moscovich, Yoel Shkolnisky, and Amit Singer. Computationalcryoem/aspire-python: v0.14.2, December 2025.

Ellen D. Zhong, Tristan Bepler, Joseph H. Davis, and Bonnie Berger. Reconstructing continuous distributions of 3D protein structure from cryo-EM images. In *International Conference on Learning Representations*, 2020.

Ellen D Zhong, Tristan Bepler, Bonnie Berger, and Joseph H Davis. CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks. *Nature methods*, 18(2):176–185, 2021a.

Ellen D Zhong, Adam Lerer, Joseph H Davis, and Bonnie Berger. CryoDRGN2: Ab initio neural reconstruction of 3d protein structures from real cryo-EM images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4066–4075, 2021b.

## A Related work on unsupervised denoising

Denoising without access to clean ground-truth images for supervised training has received substantial attention. Representative methods include Noise2Noise (Lehtinen et al., 2018), Noise2Void (Krull et al., 2019), Noisier2Noise (Moran et al., 2020), SURE (Soltanayev & Chun, 2018), and Noise2Score (Kim & Ye, 2021). Several cryo-EM methods and software packages use Noise2Noise-style denoisers to enhance micrograph contrast and facilitate visual inspection or particle picking (Tegunov & Cramer, 2019; Bepler et al., 2020; Punjani et al., 2017; Wagner & Raunser, 2020; Structura Biotechnology Inc., 2026). **Noise2Noise** requires pairs of conditionally independent noisy measurements corresponding to the same latent signal, which is a demanding requirement in many practical settings. A motion-corrected cryo-EM micrograph is obtained by aligning and averaging multiple movie frames. Two Noise2Noise inputs can therefore be formed by splitting the frames into non-overlapping subsets, such as odd- and even-indexed frames, and averaging each subset into a separate half-micrograph. The resulting denoised micrographs have higher visual contrast and can aid particle picking, but they do not contain increased experimental signal for subsequent 2D or 3D reconstruction (Structura Biotechnology Inc., 2026). To our knowledge, **Noisier2Noise** has not previously been combined with score distillation to jointly denoise and correct the CTF of extracted single-particle cryo-EM images. Noisier2Noise fits naturally into our fully score-based setup because the score model can be trained directly on noisy observations. It adds an independent perturbation to already noisy data and uses the score of the resulting mollified distribution to estimate the noise-free signal. The score model learned from the raw particle stack provides a noise-free but still CTF-affected target, which is coupled to an SDS-like optimization over the unknown CTF-free particle image (see Section 3.1).

## B Algorithms

**Algorithm 1:** Anchor-Conditioned 3D Reconstruction

**Input:** denoiser $D_{\boldsymbol{\theta}}$, anchor image $\mathbf{x}^i_{\text{clean}}$ (shape size $\times$ size), num. of steps $S$, batch size $M > 1$, mollification degree $t$, learning rate $\eta > 0$, anchor weight $w_a(k)$, score weight $w_s(k)$
$\mathbf{V}_i[x, y, z] \leftarrow \text{mean}\left(\mathbf{x}^i_{\text{clean}}\right)/\text{size} \quad \forall\, x, y, z \in \{1, \ldots, \text{size}\}$ ▷ *initialize volume*
**for** $k \in 1, \ldots, S$ **do**
  $\mathbf{R}_m \sim p_k(\mathbf{R}) \quad \forall\, m \in \{2, \ldots, M\}$ ▷ *forward projection*
  $\mathbf{R}_1 \leftarrow \mathbf{I}$
  $\mathbf{x}_m \leftarrow \mathbf{f}(\mathbf{V}_i, \mathbf{R}_m) \quad \forall\, m \in \{1, \ldots, M\}$
  $\boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \forall\, m \in \{2, \ldots, M\}$ ▷ *compute residuals*
  $\mathbf{r}_m \leftarrow w_s(k)(\mathbf{x}_m - D_{\boldsymbol{\theta}}(\mathbf{x}_m + t\boldsymbol{\epsilon}_m, t)) \quad \forall\, m \in \{2, \ldots, M\}$
  $\mathbf{r}_1 \leftarrow w_a(k)(\mathbf{x}_1 - \mathbf{x}^i_{\text{clean}})$
  $\Delta\mathbf{V}_i \leftarrow \texttt{backproject}(\mathbf{r}_m, \mathbf{R}_m)$ ▷ *back project residuals*
  $\mathbf{V}_i \leftarrow \mathbf{V}_i - \eta\, \Delta\mathbf{V}_i$ ▷ *update volume*
**end for**
**Return:** volume $\mathbf{V}_i$ corresponding to $\mathbf{x}^i_{\text{raw}}$ and $\mathbf{x}^i_{\text{clean}}$

# C PROOFS

## C.1 GRADIENT OF THE JOINT DENOISING AND CTF-CORRECTION LOSS

Here we prove that the gradient of Equation 5 is Equation 6. Let $\pi_j = 1/N$ denote the uniform probability of drawing CTF row $j$, and define

$$\boldsymbol{\mu}_j = \mathcal{A}(\mathbf{x}^i_{\text{clean}}, \text{CTF}_j), \qquad \mathbf{J}_j = \frac{\partial \mathcal{A}(\mathbf{x}^i_{\text{clean}}, \text{CTF}_j)}{\partial \mathbf{x}^i_{\text{clean}}}. \tag{10}$$

We reparameterize samples from $q^{i,j}_\tau = \mathcal{N}(\boldsymbol{\mu}_j, \tau^2\mathbf{C})$ as

$$\mathbf{z}^{i,j}_{\alpha\sigma} = \boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}), \qquad \tau^2 = (1+\alpha^2)\sigma^2. \tag{11}$$

Since $p^{\boldsymbol{\theta}}_{\text{ctf},\tau} \equiv p^{\boldsymbol{\theta}}_{\text{raw},\alpha\sigma}$, the denoiser evaluated at timescale $\alpha\sigma$ satisfies

$$\nabla_{\mathbf{z}^{i,j}_{\alpha\sigma}} \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\mathbf{z}^{i,j}_{\alpha\sigma}\right) = \left(\alpha^2\sigma^2\mathbf{C}\right)^{-1}\left[D_{\boldsymbol{\theta}}\left(\mathbf{z}^{i,j}_{\alpha\sigma}, \alpha\sigma\right) - \mathbf{z}^{i,j}_{\alpha\sigma}\right]. \tag{12}$$

The corresponding score-based estimate of the noise-free, CTF-affected image is

$$\mathbf{z}^{i,j}_{\alpha\sigma} + \tau^2\mathbf{C}\nabla_{\mathbf{z}^{i,j}_{\alpha\sigma}} \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\mathbf{z}^{i,j}_{\alpha\sigma}\right) = \frac{(1+\alpha^2)D_{\boldsymbol{\theta}}\left(\mathbf{z}^{i,j}_{\alpha\sigma}, \alpha\sigma\right) - \mathbf{z}^{i,j}_{\alpha\sigma}}{\alpha^2}. \tag{13}$$

In this equality we can clearly see the connection to the Noisier2Noise estimator proposed by Moran et al. (2020). We begin with the score contribution to Equation 6 and show that it is the gradient of the reverse-KL term in Equation 5:

$$\begin{aligned}
\mathbf{g}^i_{\text{score}} &:= -w_s\tau^2\mathbb{E}_{\substack{j\sim\pi \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{C})}}\left[\mathbf{J}_j^\top \nabla \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right)\right] && (14)\\
&= w_s\mathbb{E}_{\substack{j\sim\pi \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{C})}}\left[\mathbf{J}_j^\top\mathbf{C}^{-1}\left(\boldsymbol{\mu}_j - \mathbf{z}^{i,j}_{\alpha\sigma} - \tau^2\mathbf{C}\nabla_{\mathbf{z}^{i,j}_{\alpha\sigma}} \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\mathbf{z}^{i,j}_{\alpha\sigma}\right)\right)\right] && (15)\\
&= -w_s\mathbb{E}_{\substack{j\sim\pi \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{C})}}\left[\mathbf{J}_j^\top\left(\tau^2\nabla \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right) + \tau\mathbf{C}^{-1}\boldsymbol{\epsilon}\right)\right] && (16)\\
&= -w_s\nabla_{\mathbf{x}^i_{\text{clean}}}\mathbb{E}_{\substack{j\sim\pi \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{C})}}\left[\tau^2 \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right) + \tau\boldsymbol{\epsilon}^\top\mathbf{C}^{-1}\boldsymbol{\mu}_j\right] && (17)\\
&= -w_s\nabla_{\mathbf{x}^i_{\text{clean}}}\left(\tau^2\mathbb{E}_{\substack{j\sim\pi \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{C})}}\left[\log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right)\right] + \tau\underbrace{\mathbb{E}_{\substack{j\sim\pi \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{C})}}\left[\boldsymbol{\epsilon}^\top\mathbf{C}^{-1}\boldsymbol{\mu}_j\right]}_{0}\right) && (18)\\
&= -w_s\tau^2\nabla_{\mathbf{x}^i_{\text{clean}}}\sum_{j=1}^N \pi_j \int_{\mathbb{R}^d} \mathcal{N}\left(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{C}\right) \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right)\, \mathrm{d}\boldsymbol{\epsilon} && (19)\\
&= -w_s\tau^2\nabla_{\mathbf{x}^i_{\text{clean}}}\sum_{j=1}^N \pi_j \int_{\mathbb{R}^d} \mathcal{N}\left(\mathbf{y}; \boldsymbol{\mu}_j, \tau^2\mathbf{C}\right) \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\mathbf{y}\right)\, \mathrm{d}\mathbf{y} && (20)\\
&= w_s\tau^2\nabla_{\mathbf{x}^i_{\text{clean}}}\Bigg(\underbrace{\sum_{j=1}^N \pi_j \int_{\mathbb{R}^d} \mathcal{N}\left(\mathbf{y}; \boldsymbol{\mu}_j, \tau^2\mathbf{C}\right) \log \mathcal{N}\left(\mathbf{y}; \boldsymbol{\mu}_j, \tau^2\mathbf{C}\right)\, \mathrm{d}\mathbf{y}}_{\text{constant in } \mathbf{x}^i_{\text{clean}}} \\
&\qquad\qquad - \sum_{j=1}^N \pi_j \int_{\mathbb{R}^d} \mathcal{N}\left(\mathbf{y}; \boldsymbol{\mu}_j, \tau^2\mathbf{C}\right) \log p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\left(\mathbf{y}\right)\, \mathrm{d}\mathbf{y}\Bigg) && (21)\\
&= w_s\tau^2\nabla_{\mathbf{x}^i_{\text{clean}}}\sum_{j=1}^N \pi_j D_{\text{KL}}\left(\mathcal{N}\left(\boldsymbol{\mu}_j, \tau^2\mathbf{C}\right) \,\middle\|\, p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\right) && (22)\\
&= \nabla_{\mathbf{x}^i_{\text{clean}}}\left\{w_s\tau^2\mathbb{E}_j\left[D_{\text{KL}}\left(q^{i,j}_\tau \,\middle\|\, p^{\boldsymbol{\theta}}_{\text{ctf},\tau}\right)\right]\right\}. && (23)
\end{aligned}$$

Here, $d$ is the number of pixels in image space. The individual transformations are as follows:

1. Equation 14 → Equation 15: substitute the score-based target from Equation 13. Because $\mathbf{z}^{i,j}_{\alpha\sigma} = \boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}$, we have

$$\boldsymbol{\mu}_j - \mathbf{z}^{i,j}_{\alpha\sigma} - \tau^2\mathbf{C}\nabla_{\mathbf{z}^{i,j}_{\alpha\sigma}} \log p^{\boldsymbol{\theta}}_{\mathrm{ctf},\tau}\left(\mathbf{z}^{i,j}_{\alpha\sigma}\right) = -\tau\boldsymbol{\epsilon} - \tau^2\mathbf{C}\nabla \log p^{\boldsymbol{\theta}}_{\mathrm{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right). \tag{24}$$

   After multiplication by $\mathbf{C}^{-1}$, the noise contribution has zero expectation.

2. Equation 15 → Equation 16: expand the residual using the identity above.

3. Equation 16 → Equation 17: interchange the gradient and expectation and apply the chain-rule identities

$$\nabla_{\mathbf{x}^i_{\mathrm{clean}}} \log p^{\boldsymbol{\theta}}_{\mathrm{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right) = \mathbf{J}_j^\top \nabla \log p^{\boldsymbol{\theta}}_{\mathrm{ctf},\tau}\left(\boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}\right), \tag{25}$$

$$\nabla_{\mathbf{x}^i_{\mathrm{clean}}} \left(\boldsymbol{\epsilon}^\top\mathbf{C}^{-1}\boldsymbol{\mu}_j\right) = \mathbf{J}_j^\top\mathbf{C}^{-1}\boldsymbol{\epsilon}. \tag{26}$$

4. Equation 17 → Equation 18: use linearity of expectation. Since $\boldsymbol{\epsilon}$ is zero mean and independent of $\boldsymbol{\mu}_j$,

$$\mathbb{E}\left[\boldsymbol{\epsilon}^\top\mathbf{C}^{-1}\boldsymbol{\mu}_j\right] = 0. \tag{27}$$

5. Equation 18 → Equation 19: write the expectation as a uniform sum over CTF rows and an integral over the colored Gaussian perturbation.

6. Equation 19 → Equation 20: substitute $\mathbf{y} = \boldsymbol{\mu}_j + \tau\boldsymbol{\epsilon}$. Consequently, $\boldsymbol{\epsilon} = (\mathbf{y} - \boldsymbol{\mu}_j)/\tau$, $\mathrm{d}\boldsymbol{\epsilon} = \tau^{-d}\,\mathrm{d}\mathbf{y}$, and

$$\mathcal{N}\left(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{C}\right) = \tau^d \mathcal{N}\left(\mathbf{y}; \boldsymbol{\mu}_j, \tau^2\mathbf{C}\right). \tag{28}$$

7. Equation 20 → Equation 21: add the negative entropy of the fixed-covariance Gaussian. This quantity is independent of its mean and therefore independent of $\mathbf{x}^i_{\mathrm{clean}}$:

$$\int_{\mathbb{R}^d} \mathcal{N}\left(\mathbf{y}; \boldsymbol{\mu}_j, \tau^2\mathbf{C}\right) \log \mathcal{N}\left(\mathbf{y}; \boldsymbol{\mu}_j, \tau^2\mathbf{C}\right)\, \mathrm{d}\mathbf{y} = -\frac{1}{2}\left[d\log(2\pi e) + \log\det(\tau^2\mathbf{C})\right]. \tag{29}$$

8. Equation 21 → Equation 22: apply the definition of the KL divergence.

9. Equation 22 → Equation 23: use $q^{i,j}_\tau = \mathcal{N}(\boldsymbol{\mu}_j, \tau^2\mathbf{C})$ and the definition of the expectation over uniformly sampled CTF rows.

It remains to consider the particle-specific anchor. Let $\mathbf{x}^i_{\mathrm{ctf}}$ denote the score-based denoising estimate obtained from

$$\mathbf{z}^i_{\alpha\sigma} = \mathbf{x}^i_{\mathrm{raw}} + \alpha\sigma\boldsymbol{\epsilon}_a, \qquad \boldsymbol{\epsilon}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{C}), \tag{30}$$

using the score-based estimator in Equation 13. This estimate is held fixed while optimizing $\mathbf{x}^i_{\mathrm{clean}}$. The anchor contribution to Equation 6 is therefore

$$\begin{aligned}\mathbf{g}^i_{\mathrm{anchor}} &:= w_a(\mathbf{J}^{i,i}_{\mathcal{A}})^\top\left[\mathcal{A}(\mathbf{x}^i_{\mathrm{clean}}, \mathrm{CTF}_i) - \mathbf{x}^i_{\mathrm{ctf}}\right] \\ &= \nabla_{\mathbf{x}^i_{\mathrm{clean}}}\left\{\frac{w_a}{2}\left\|\mathcal{A}(\mathbf{x}^i_{\mathrm{clean}}, \mathrm{CTF}_i) - \mathbf{x}^i_{\mathrm{ctf}}\right\|_2^2\right\}.\end{aligned} \tag{31}$$

Combining Equations 23 and 31 gives

$$\begin{aligned}\nabla_{\mathbf{x}^i_{\mathrm{clean}}}\mathcal{L}^i_{\mathrm{2D}} &= \mathbf{g}^i_{\mathrm{score}} + \mathbf{g}^i_{\mathrm{anchor}} \\ &= \nabla_{\mathbf{x}^i_{\mathrm{clean}}}\left\{w_s\tau^2\mathbb{E}_j\left[D_{\mathrm{KL}}\left(q^{i,j}_\tau \,\middle\|\, p^{\boldsymbol{\theta}}_{\mathrm{ctf},\tau}\right)\right] + \frac{w_a}{2}\left\|\mathcal{A}(\mathbf{x}^i_{\mathrm{clean}}, \mathrm{CTF}_i) - \mathbf{x}^i_{\mathrm{ctf}}\right\|_2^2\right\}.\end{aligned} \tag{32}$$

The expression inside the braces is precisely $\mathcal{L}^i_{\mathrm{2D}}$ from Equation 5, which proves Equation 6.

### C.2 Gradient of the anchor-conditioned reconstruction objective

In this section, we prove that the gradient of Equation 8 is Equation 9. The derivation of the score term follows the construction of Equation (4) in Poole et al. (2022). For compactness, define $\boldsymbol{\mu}_{\mathbf{R}} = \mathbf{f}(\mathbf{V}_i, \mathbf{R})$ and $\mathbf{J}_{\mathbf{R}} = \partial\mathbf{f}(\mathbf{V}_i, \mathbf{R})/\partial\mathbf{V}_i$. The denoiser satisfies $D_{\boldsymbol{\theta}}(\mathbf{y}, t) = \mathbf{y} + t^2\nabla_{\mathbf{y}} \log p^{\boldsymbol{\theta}}_{\mathrm{clean},t}(\mathbf{y})$

(Karras et al., 2022). For any reconstruction rotation distribution $\rho$ that is independent of $\mathbf{V}_i$, the score-gradient estimator can therefore be written as

$$\mathbb{E}_{\substack{\mathbf{R}\sim\rho \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[\mathbf{J}_{\mathbf{R}}^{\top}\left(\boldsymbol{\mu}_{\mathbf{R}} - D_{\boldsymbol{\theta}}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon}, t)\right)\right] \tag{33}$$

$$= -\mathbb{E}_{\substack{\mathbf{R}\sim\rho \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[\mathbf{J}_{\mathbf{R}}^{\top}\left(t^2 \nabla \log p^{\boldsymbol{\theta}}_{\text{clean},t}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon}) + t\boldsymbol{\epsilon}\right)\right] \tag{34}$$

$$= -\nabla_{\mathbf{V}_i} \mathbb{E}_{\substack{\mathbf{R}\sim\rho \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[t^2 \log p^{\boldsymbol{\theta}}_{\text{clean},t}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon}) + t\boldsymbol{\epsilon}^{\top}\boldsymbol{\mu}_{\mathbf{R}}\right] \tag{35}$$

$$= -\nabla_{\mathbf{V}_i} \left( t^2 \mathbb{E}_{\substack{\mathbf{R}\sim\rho \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[\log p^{\boldsymbol{\theta}}_{\text{clean},t}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon})\right] + t \underbrace{\mathbb{E}_{\substack{\mathbf{R}\sim\rho \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[\boldsymbol{\epsilon}^{\top}\boldsymbol{\mu}_{\mathbf{R}}\right]}_{0} \right) \tag{36}$$

$$= -\nabla_{\mathbf{V}_i}\, t^2 \mathbb{E}_{\substack{\mathbf{R}\sim\rho \\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[\log p^{\boldsymbol{\theta}}_{\text{clean},t}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon})\right] \tag{37}$$

$$= -\nabla_{\mathbf{V}_i}\, t^2 \int_{\mathrm{SO}(3)} \int_{\mathbb{R}^d} \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I}) \log p^{\boldsymbol{\theta}}_{\text{clean},t}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon})\, \mathrm{d}\boldsymbol{\epsilon}\, \mathrm{d}\rho(\mathbf{R}) \tag{38}$$

$$= -\nabla_{\mathbf{V}_i}\, t^2 \int_{\mathrm{SO}(3)} \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I}) \log p^{\boldsymbol{\theta}}_{\text{clean},t}(\mathbf{y})\, \mathrm{d}\mathbf{y}\, \mathrm{d}\rho(\mathbf{R}) \tag{39}$$

$$= \nabla_{\mathbf{V}_i}\, t^2 \Bigg( \underbrace{\int_{\mathrm{SO}(3)} \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I}) \log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I})\, \mathrm{d}\mathbf{y}\, \mathrm{d}\rho(\mathbf{R})}_{\text{constant in } \mathbf{V}_i} - \int_{\mathrm{SO}(3)} \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I}) \log p^{\boldsymbol{\theta}}_{\text{clean},t}(\mathbf{y})\, \mathrm{d}\mathbf{y}\, \mathrm{d}\rho(\mathbf{R}) \Bigg) \tag{40}$$

$$= \nabla_{\mathbf{V}_i}\, t^2 \int_{\mathrm{SO}(3)} D_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I}) \,\|\, p^{\boldsymbol{\theta}}_{\text{clean},t}\right) \mathrm{d}\rho(\mathbf{R}). \tag{41}$$

Here $d$ is the number of pixels in image space. The individual steps use the following transformations:

1. Equation 33 $\to$ Equation 34: substitute the denoiser identity above.
2. Equation 34 $\to$ Equation 35: interchange the gradient and expectation and apply the chain-rule identities
$$\nabla_{\mathbf{V}_i} \log p^{\boldsymbol{\theta}}_{\text{clean},t}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon}) = \mathbf{J}_{\mathbf{R}}^{\top} \nabla \log p^{\boldsymbol{\theta}}_{\text{clean},t}(\boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon}), \tag{42}$$
$$\nabla_{\mathbf{V}_i} \boldsymbol{\epsilon}^{\top}\boldsymbol{\mu}_{\mathbf{R}} = \mathbf{J}_{\mathbf{R}}^{\top}\boldsymbol{\epsilon}. \tag{43}$$
3. Equation 35 $\to$ Equation 36: use linearity of expectation.
4. Equation 36 $\to$ Equation 37: since $\boldsymbol{\epsilon}$ is zero mean and independent of $\boldsymbol{\mu}_{\mathbf{R}}$, $\mathbb{E}[\boldsymbol{\epsilon}^{\top}\boldsymbol{\mu}_{\mathbf{R}}] = 0$.
5. Equation 37 $\to$ Equation 38: write the expectation as an integral with respect to the Gaussian noise and the rotation probability measure $\rho$.
6. Equation 38 $\to$ Equation 39: substitute $\mathbf{y} = \boldsymbol{\mu}_{\mathbf{R}} + t\boldsymbol{\epsilon}$, so that $\boldsymbol{\epsilon} = (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{R}})/t$, $\mathrm{d}\boldsymbol{\epsilon} = t^{-d}\,\mathrm{d}\mathbf{y}$, and
$$\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I}) = t^d \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I}). \tag{44}$$
7. Equation 39 $\to$ Equation 40: add the negative entropy of the fixed-covariance Gaussian. It is independent of its mean and hence of $\mathbf{V}_i$:
$$\int_{\mathbb{R}^d} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I}) \log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{R}}, t^2\mathbf{I})\, \mathrm{d}\mathbf{y} = -\frac{d}{2}\left(\log(2\pi t^2) + 1\right). \tag{45}$$
8. Equation 40 $\to$ Equation 41: apply the definition of the KL divergence.

Thus, multiplying Equation 33 by $w_s$ gives the gradient of the first term of Equation 8 for either the uniform or progressive rotation distribution.

Table 1: Overview of score-model training, denoising and CTF correction, and reconstruction configurations for each dataset. Score-model training and 3D reconstruction were performed on a single NVIDIA A100 GPU with 80 GB of VRAM. Denoising and CTF correction used eight GPUs in parallel. We report the corresponding aggregate GPU time as an equivalent single-GPU runtime. Both score models were trained with mixed precision using Adam with its default parameters.

| **Setting** | **IgG-RL** | **IgG-1D** | **capsidHA** | **EMPIAR-10295** |
|---|---|---|---|---|
| *Raw-particle score-model training* | | | | |
| Image size | 96 | 96 | 104 | 128 |
| Learning rate | $2 \times 10^{-4}$ | $2 \times 10^{-4}$ | $2 \times 10^{-4}$ | $2 \times 10^{-4}$ |
| $t_{\min}$ | 0.002 | 0.002 | 0.002 | 0.002 |
| $t_{\max}$ | 2 | 2 | 2 | 2 |
| Batch size | 128 | 128 | 128 | 128 |
| Training steps | $\sim$ 116k | $\sim$ 116k | $\sim$ 97k | $\sim$ 64k |
| Runtime | $\sim$ 10.8 h | $\sim$ 10.8 h | $\sim$ 10.8 h | $\sim$ 10.7 h |
| *Denoising and CTF correction* | | | | |
| Data-noise scale $\sigma$ | 0.98 | 0.99 | 0.98 | 0.8 |
| Augmentation multiplier $\alpha$ | 0.25 | 0.25 | 0.25 | 0.25 |
| CTF samples $M$ | 1 | 1 | 1 | 1 |
| Optimization steps $S$ | 25 | 25 | 25 | 50 |
| Learning rate $\beta$ | 0.5 | 0.5 | 0.5 | 0.5 |
| Score weight $w_s$ | 1 | 1 | 1 | 1 |
| Anchor weight $w_a$ | 1 | 1 | 1 | 1 |
| Wiener parameter $\gamma$ | 0.001 | 0.001 | 0.001 | 0.001 |
| Runtime per image | $\sim$ 0.36 s | $\sim$ 0.36 s | $\sim$ 0.5 s | $\sim$ 1.5 s |
| *Clean-image score-model training* | | | | |
| Image size | 96 | 96 | 104 | 112 |
| Learning rate | $2 \times 10^{-4}$ | $2 \times 10^{-4}$ | $2 \times 10^{-4}$ | $2 \times 10^{-4}$ |
| $t_{\min}$ | 0.002 | 0.002 | 0.002 | 0.002 |
| $t_{\max}$ | 4 | 4 | 4 | 4 |
| Batch size | 128 | 128 | 128 | 128 |
| Training steps | $\sim$ 139k | $\sim$ 139k | $\sim$ 175k | $\sim$ 416k |
| Runtime | $\sim$ 13 h | $\sim$ 13 h | $\sim$ 19.5 h | $\sim$ 53.9 h |
| *Anchor-conditioned 3D reconstruction* | | | | |
| Projections per step $M$ | 2 | 2 | 2 | 2 |
| Mollification degree $t$ | 2.5 | 2.5 | 1.5 | 0.5 |
| Learning rate $\eta$ | 1 | 3 | 3 | 3 |
| Score weight $w_s(k)$ | 1 | 1 | 1 | 1 |
| Anchor weight $w_a(k)$ | 1 | 1 | 1 | $1 - k/S$ |
| Reconstruction steps $S$ | 10k | 10k | 10k | 10k |
| Rotation sampling | $\mathcal{U}(\mathrm{SO}(3))$ | $\mathcal{U}(\mathrm{SO}(3))$ | $p : 0.3 \to 0.6$ | $p : 0.3 \to 0.6$ |
| Runtime per volume | $\sim$ 1.9 min | $\sim$ 1.9 min | $\sim$ 1.96 min | $\sim$ 1.97 min |

The anchor term is an ordinary least-squares loss

$$\mathcal{J}_{\mathrm{anchor}}(\mathbf{V}_i) = \frac{w_a}{2} \left\| \mathbf{f}(\mathbf{V}_i, \mathbf{I}) - \mathbf{x}^i_{\mathrm{clean}} \right\|_2^2, \tag{46}$$

with gradient

$$\nabla_{\mathbf{V}_i} \mathcal{J}_{\mathrm{anchor}}(\mathbf{V}_i) = w_a \mathbf{J}(\mathbf{V}_i, \mathbf{I})^\top \left( \mathbf{f}(\mathbf{V}_i, \mathbf{I}) - \mathbf{x}^i_{\mathrm{clean}} \right). \tag{47}$$

Adding this anchor gradient to $w_s$ times Equation 33 gives Equation 9.

# D EXPERIMENTAL DETAILS

## D.1 CAPSIDHA SIMULATION

**Conformational model.** capsidHA is a synthetic continuously heterogeneous model with a static icosahedral wireframe core and an arm at each of its twelve vertices. Every arm consists of a cylindrical shaft and a spherical terminal knob. The six vertices in one hemisphere tilt tangentially toward the positive symmetry axis, while the other six follow the mirrored motion; therefore, a single angle moves the complete structure continuously between closed and open configurations. We sampled 200 distinct angles independently and uniformly between the fully-closed and fully-open conformations. The increased shaft and knob radii give the moving arms substantially more mass than the icosahedral core, making the heterogeneous signal less dominated by the static core.

**Particle and CTF simulation.** Using ASPIRE (Wright et al., 2025), we generated $25{,}000$ projections at $104 \times 104$ pixels and $3$ Å/pixel from randomly selected conformations and viewing directions. Independent in-plane shifts were sampled from a zero-mean Gaussian with standard deviation of two pixels in each coordinate, and all particle amplitudes were set to one. We used CTFs with defocus values linearly spaced from $10{,}000$ to $15{,}000$ Å. The accelerating voltage was set to $300$ kV, the spherical aberration to $2.26$ mm, and the amplitude contrast to $0.1$. The B-factor was set to zero, with no astigmatism or phase shift applied. The generating conformation of every particle image was retained for evaluation.

**Empirical colored-noise distribution.** To model noise for the capsidHA dataset in a highly realistic way, we used a colored noise distribution whose spatial correlations are estimated from particle-free regions of the experimental EMPIAR-10454 yeast fatty-acid-synthase dataset. The background regions of this dataset show strong, visually apparent spatial correlations; here, colored noise means that its power is distributed non-uniformly over spatial frequency. We manually selected background crops that did not contain a visible particle signal. For each crop $\mathbf{b}_k$, we subtracted its mean, zero-padded it to the $208 \times 208$ spectral grid, computed its orthonormal periodogram, and normalized that periodogram to unit mean power. We then averaged the normalized spectra over crops, averaged over radial frequency rings, and normalized the result once more:

$$S_{10454}(\boldsymbol{\omega}) = \mathrm{radialize}\left[\frac{1}{B}\sum_{k=1}^{B}\frac{\left|\mathcal{F}\{\mathrm{pad}(\mathbf{b}_k - \bar{b}_k)\}(\boldsymbol{\omega})\right|^2}{\mathrm{mean}_{\boldsymbol{\omega}'}\left|\mathcal{F}\{\mathrm{pad}(\mathbf{b}_k - \bar{b}_k)\}(\boldsymbol{\omega}')\right|^2}\right]. \tag{48}$$

The background data have a pixel spacing of $2.65$ Å, whereas capsidHA uses $3$ Å. We therefore evaluate the estimated profile at $(2.65/3.0)\|\boldsymbol{\omega}\|$, which preserves its physical correlation length in Angstroms rather than its width in pixels. Finally, we scale the discrete mean power to a target real-space variance and sample the colored noise with ASPIRE's `CustomNoiseAdder`. The realized noise variance is $4.9992 \times 10^{-4}$ and the variance of the CTF-affected noise-free stack is $3.0843 \times 10^{-5}$, giving $\mathrm{SNR} = 0.0617$.

This value is not directly comparable to the nominal SNRs reported by CryoBench (Jeon et al., 2024), including $\mathrm{SNR} = 0.01$ for IgG-RL and $\mathrm{SNR} = 0.005$ for IgG-1D. The CryoBench simulation software `cryosim` defines SNR using global pixel variance over the complete boxed image, including large regions containing only noise, and reports this value before subsequent Fourier cropping. For example, the nominal IgG-RL SNR of $0.01$ refers to the $256^2$ images before Fourier cropping to $128^2$. For white noise, halving the retained Fourier width along both axes preserves approximately one quarter of the noise power. If the relevant signal power is retained, the corresponding SNR therefore increases by a factor of four, from $0.01$ to approximately $0.04$. Expressing the same particle signal in a for IgG sufficiently large $88^2$ box instead of a $128^2$ box removes additional noise-only support and introduces a further factor of $128^2/88^2 \approx 2.116$. The resulting box-matched white-noise reference is, therefore, approximately

$$0.01 \times 4 \times \frac{128^2}{88^2} \approx 0.0846.$$

This calculation illustrates how strongly the nominal SNR depends on the processing stage and the amount of signal-free support. Also, white noise has a flat power spectrum, so reducing the Fourier area to one quarter also removes approximately three quarters of its power. Cryo-EM background noise like that in EMPIAR-10454 instead contains substantial low-frequency power. Fourier cropping

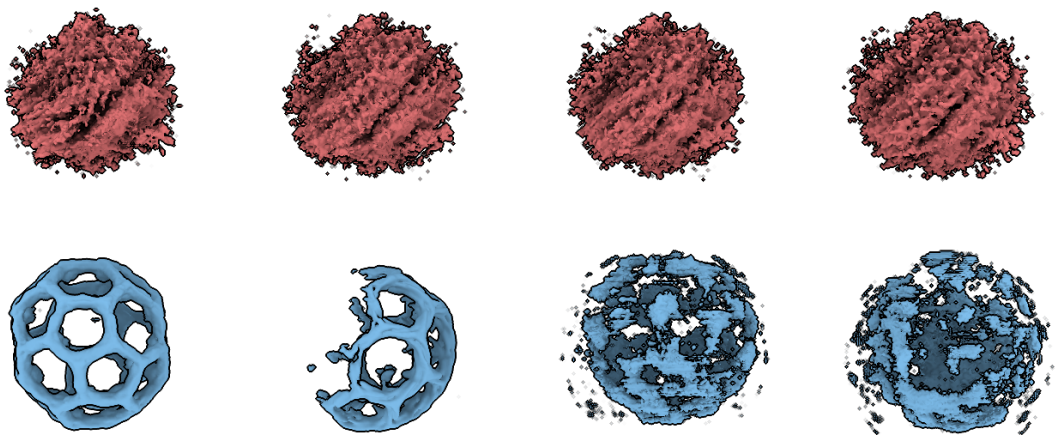


Figure 6: Representative failure cases on EMPIAR-10295. **Top:** Four randomly selected CryoDRGN-AI reconstructions (red). The reconstructed volumes collapse to similar approximately spherical densities and do not recover the distinct clathrin-cage architectures. **Bottom:** Four unsuccessful CryoSD reconstructions (blue). From left to right: (1) the cage topology is recovered, but the triskelia are strongly oversmoothed; (2) individual triskelia contain more detail, but the global cage is broken and incomplete; and (3, 4) optimization failures in which the reconstruction becomes trapped in a poor local optimum.

retains this central low-frequency region and can therefore preserve considerably more than one quarter of the colored-noise power. Its effective SNR consequently increases less rapidly under Fourier cropping than the white-noise calculation suggests.

### D.2 EMPIAR-10295 EXPERIMENTAL DETAILS

**Dataset and preprocessing.** EMPIAR-10295 (Morris et al., 2019) contains particle images associated with five clathrin-cage structures: the 28-triskelion mini coat (EMD-0114), 32-triskelion sweet potato (EMD-0115), 36-triskelion D6 barrel (EMD-0116), 36-triskelion tennis ball (EMD-0118), and 37-triskelion big apple (EMD-0120). The deposited particles are already separated by structure across multiple MRC stacks and corresponding STAR files. For our experiment, we concatenated the component stacks back into a single particle stack and concatenated the STAR rows in exactly the same order, thereby preserving the particle-to-CTF correspondence while withholding the deposited structure labels from CryoSD, RELION and CryoDRGN-AI. The 9500 deposited images have size $750 \times 750$ pixels and pixel spacing $1.705$ Å/pixel. Several images showed notable edge artifacts because the particles were cropped too close to the edge of the micrograph. We filtered them out and continued with the remaining 8183 raw particle images. We downsampled them by a factor of five to $150 \times 150$ pixels and $8.525$ Å/pixel, preserving the physical field of view, and then center-cropped them to $128 \times 128$ pixels for restoration. After denoising and CTF correction, we center-cropped the particles to $112 \times 112$ pixels for reconstruction. Cropping does not alter the pixel spacing, so the final box width is $112 \cdot 8.525 = 954.8$ Å.

**Big apple topology.** The big apple comparison exposes a limitation of treating the deposited map as ground truth. The theoretical big apple reference used by Morris et al. (2019) contained 38 triskelia, whereas their supervised classification produced a volume containing 37; this difference is stated explicitly in the original study, and the resulting 37-triskelion density was deposited as EMD-0120. Our selected *ab initio* reconstruction is instead consistent with the complete theoretical 38-triskelion cage. A topological analysis of its density-derived marker network constructed using ChimeraX (Meng et al., 2023) gives 38 vertices and 57 edges, with degree three at every vertex. The network cont ains exactly 12 pentagonal and 9 hexagonal faces, with 11 pentagon-pentagon, 38 pentagon-hexagon, and 8 hexagon-hexagon shared edges. This matches the expected theoretical big apple topology.