

Figure 3: Unconditional samples from the diffusion prior trained on the ShapeNet-Chair dataset. Sampled with Algorithm 1 using  $\beta(t) = 1/t$  if  $t > 1$  else 1,  $t_{\max} = 80$  and 100 time steps. The images are created using the surface mode of PyMOL (Schrödinger & DeLano, 2020).

## A APPENDIX

### A.1 RECONSTRUCTION GUIDANCE

For completeness we show how to derive the approximation  $p_t(\mathbf{y} | \mathbf{x}(t)) \approx p_0(\mathbf{y} | \mathbf{D}_\theta(\mathbf{x}(t), t))$  used in reconstruction guidance to perform approximate diffusion posterior sampling. To take advantage of the likelihood on the noiseless data  $p_0(\mathbf{x}(0) | \mathbf{y})$ , we follow the argument of Chung et al. (2023) and first express  $p_t(\mathbf{y} | \mathbf{x}(t))$  as the marginal

$$p_t(\mathbf{y} | \mathbf{x}(t)) = \int p(\mathbf{y} | \mathbf{x}(t), \mathbf{x}(0)) p(\mathbf{x}(0) | \mathbf{x}(t)) d\mathbf{x}(0). \quad (15)$$

Given  $\mathbf{x}(0)$ ,  $\mathbf{y}$  is independent of  $\mathbf{x}(t)$ . Therefore, we can simplify  $p(\mathbf{y} | \mathbf{x}(t), \mathbf{x}(0))$  to  $p_0(\mathbf{y} | \mathbf{x}(0))$  and obtain

$$p_t(\mathbf{y} | \mathbf{x}(t)) = \int p_0(\mathbf{y} | \mathbf{x}(0)) p(\mathbf{x}(0) | \mathbf{x}(t)) d\mathbf{x}(0). \quad (16)$$

Chung et al. (2023) note that  $p(\mathbf{x}(0) | \mathbf{x}(t))$  is intractable in the general case. Therefore, Chung et al. (2023) propose the approximation

$$p(\mathbf{x}(0) | \mathbf{x}(t)) \approx \delta(\mathbf{D}(\mathbf{x}(t), t) - \mathbf{x}(0)), \quad (17)$$

where  $\mathbf{D}(\mathbf{x}(t), t) := \mathbb{E}_{\mathbf{x}(0) \sim p(\cdot | \mathbf{x}(t))}[\mathbf{x}(0)]$ . The intuition behind  $\mathbf{D}$  is that it denoises the noisy input  $\mathbf{x}(t)$ . This so-called *denoising function* is in general intractable and has to be replaced by an estimator  $\mathbf{D}_\theta$  (Karras et al., 2022). Learning  $\mathbf{D}_\theta$  is an essential part of the Diffusion Model training and will be discussed in Section 3.1. By plugging (17) into (16) we obtain

$$p_t(\mathbf{y} | \mathbf{x}(t)) \approx \int p_0(\mathbf{y} | \mathbf{x}(0)) \delta(\mathbf{D}_\theta(\mathbf{x}(t), t) - \mathbf{x}(0)) d\mathbf{x}(0) = p_0(\mathbf{y} | \mathbf{D}_\theta(\mathbf{x}(t), t)). \quad (18)$$

### A.2 TRAINING

In this section, we aim to provide additional background on the objective optimized during diffusion model training and elaborate on the specifics of training our 3D diffusion priors. In diffusion model training we use gradient descent to find a good score-model  $\mathbf{s}_\theta$  via *explicit score matching* (ESM):

$$\min_{\theta} \mathbb{E}_{t, \mathbf{x}(t)} \left[ \lambda(t) \left\| \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)) - \mathbf{s}_\theta(\mathbf{x}(t), t) \right\|^2 \right] \quad (19)$$

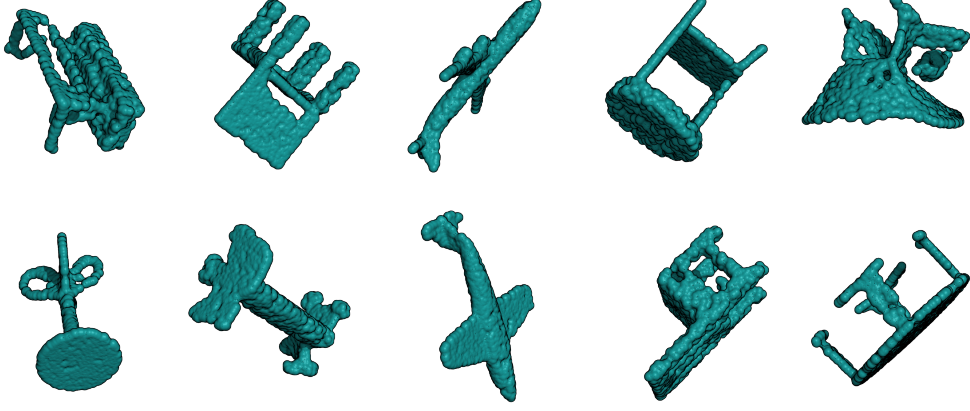


Figure 4: Unconditional samples from the diffusion prior trained on the ShapeNet-Mixed dataset. Sampled with Algorithm 1 using  $\beta(t) = 1/t$  if  $t > 1$  else 1,  $t_{\max} = 80$  and 100 time steps. The images are created using the surface mode of PyMOL (Schrödinger & DeLano, 2020).

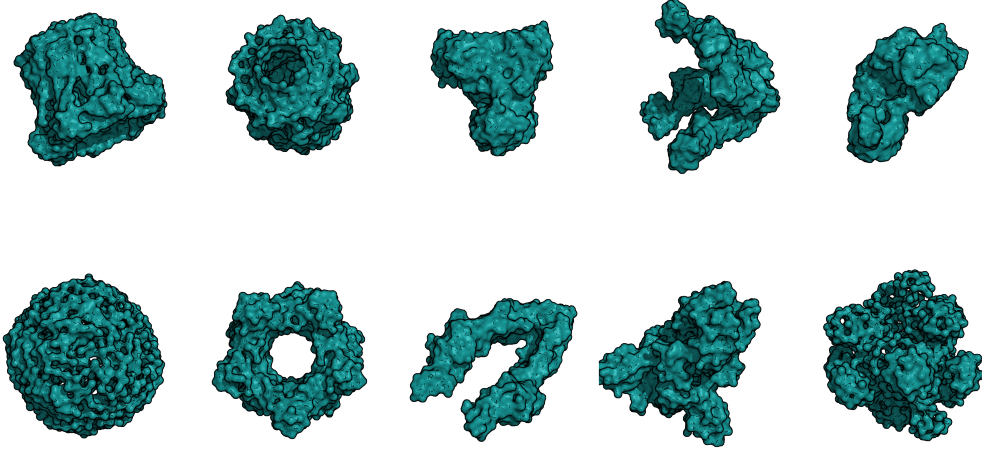


Figure 5: Unconditional samples from the diffusion prior trained on the CryoStruct dataset. Sampled with Algorithm 1 using  $\beta(t) = 1/t$  if  $t > 0.8$  else 1,  $t_{\max} = 80$  and 100 time steps. The images are created using the surface mode of PyMOL (Schrödinger & DeLano, 2020).

Table 3: All metrics (1-NNA, COV and MMD) used to quantify the diffusion prior generation performance are based on the Chamfer Distance.

Dataset	Samples from	1-NNA(%, $\downarrow$ )	COV(%, $\uparrow$ )	MMD( $[\times 10^2]$ , $\downarrow$ )
ShapeNet-Chair	Diffusion prior	78.34	44.89	27.11
	Training set	47.80	60.23	22.97
ShapeNet-Mixed	Diffusion prior	66.76	44.59	24.83
	Training set	45.91	55.41	20.63
CryoStruct	Diffusion prior	54.29	42.38	18.68
	Training set	44.94	53.05	18.13

where  $t \sim p_{\text{train}}$  (i.e.  $p_{\text{train}} = \mathcal{U}[0, T]$ ),  $\mathbf{x}(t) \sim p_t$  with the loss weighting  $\lambda : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  (i.e.  $\lambda(t) = 1/2$ ) (Hyvärinen, 2005; Vincent, 2011). Nonetheless, the score of the marginals  $\nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))$  is unknown; therefore, we do not have an explicit regression target for  $\mathbf{s}_\theta$ . Fortunately, the results from Vincent (2011) state that the optimisation problem of ESM is equivalent to the *denoising score matching* (DSM) objective:

$$\min_{\theta} \mathbb{E}_{t, \mathbf{x}(0), \mathbf{x}(t)} \left[ \lambda(t) \left\| \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) | \mathbf{x}(0)) - \mathbf{s}_\theta(\mathbf{x}(t), t) \right\|^2 \right] \quad (20)$$

where  $\mathbf{x}(0) \sim p_0$ ,  $\mathbf{x}(t) \sim p_{0t}(\cdot | \mathbf{x}(0))$ . In DSM we only need to evaluate the score of the perturbation kernel  $p_{0t}$  which is easy to calculate for suitable choices of the *drift* and *diffusion* coefficient (see for example *variance exploding* or *variance preserving* schedules in Song et al. (2021)).

We follow the design choice recommendations of Karras et al. (2022) by using  $\mathbf{f}(\mathbf{x}, t) = 0$  and  $g(t) = \sqrt{2t}$  which yields the forward diffusion SDE:  $d\mathbf{x} = \sqrt{2t} d\mathbf{w}_t$ . The resulting perturbation kernel  $p(\mathbf{x}(t) | \mathbf{x}(0))$  is a sum of the starting position  $\mathbf{x}(0)$  and infinite independent infinitesimal small Gaussian contributions. The perturbation kernel is therefore itself Gaussian:  $p_{0t}(\mathbf{x}(t) | \mathbf{x}(0)) = \mathcal{N}(\mathbf{x}(t); \mathbf{x}(0), t^2 \mathbf{I})$ . Plugging it into the DSM objective in (20) we obtain

$$\min_{\theta} \mathbb{E}_{t, \mathbf{x}(0), \mathbf{x}(t)} \left[ \lambda(t) \left\| \frac{\mathbf{x}(0) - \mathbf{x}(t)}{t^2} - \mathbf{s}_\theta(\mathbf{x}(t), t) \right\|^2 \right]. \quad (21)$$

This motivates to use the score-model parameterization:  $\mathbf{s}_\theta(\mathbf{x}(t), t) = (\mathbf{D}_\theta(\mathbf{x}(t), t) - \mathbf{x}(t))/t^2$ . Plugging it into the objective further simplifies it to

$$\min_{\theta} \mathbb{E}_{t, \mathbf{x}(0), \mathbf{x}(t)} \left[ \lambda(t) \left\| \mathbf{x}(0) - \mathbf{D}_\theta(\mathbf{x}(t), t) \right\|^2 \right], \quad (22)$$

which justifies the terminology *denoising function* for  $\mathbf{D}_\theta$ . According to Karras et al. (2022) we defined the weighting  $\lambda(t) := 1/c_{\text{out}}^2$  with  $c_{\text{out}} := 0.25t/\sqrt{t^2 + 0.25}$  and selected  $p_{\text{train}}(t)$  during training as a log-normal distribution  $\ln(t) \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$  to focus on the relevant parts of the noise schedule.

Following Karras et al. (2022), we define the denoiser by

$$\mathbf{D}_\theta(\mathbf{x}, t) := c_{\text{skip}} \mathbf{x} + c_{\text{out}}(t) F_\theta(c_{\text{in}}(t) \mathbf{x}, c_{\text{noise}}(t)) \quad (23)$$

where  $c_{\text{skip}} := 0.25/(t^2 + 0.25)$ ,  $c_{\text{in}} := 1/\sqrt{t^2 + 0.25}$  with  $c_{\text{noise}} := 1000 \cdot t$  for the ShapeNet-Chair diffusion prior and  $c_{\text{noise}} := 1000 \cdot t/t_{\text{max}}$  for the ShapeNet-Mixed and CryoStruct diffusion priors. To model  $F_\theta$ , we used the **point transformer architecture** of Nichol et al. (2022) with a width of 512 and 24 layers, where each of the 1024 points has its 3D coordinates as features. This architecture provides us with a highly flexible model with permutation equivariance. To account for the lack of rotation equivariance, we used data augmentation during training. For ShapeNet-Chair and ShapeNet-Mixed we performed **data augmentation** by transforming each point cloud during training with a random orthogonal matrix. In the case of the CryoStruct dataset, each point cloud is transformed by a random proper rotation matrix, an element of  $SO(3)$ , to avoid training on unnatural mirrored biomolecules.

To train the diffusion prior on the ShapeNet-Chair dataset, we selected  $P_{\text{mean}} = -4$ ,  $P_{\text{std}} = 1.2$  with  $t_{\text{max}} = 1$  for roughly the first 90% of the training steps and  $P_{\text{mean}} = -1.2$ ,  $P_{\text{std}} = 1.2$  with  $t_{\text{max}} = 80$  for the rest.

For the ShapeNet-Mixed and CryoStruct dataset, we selected  $P_{\text{mean}} = -2.8$ ,  $P_{\text{std}} = 0.9$  with  $t_{\text{max}} = 1$  for approximately the first 90% of the training steps and  $P_{\text{mean}} = -1.2$ ,  $P_{\text{std}} = 1.2$  with  $t_{\text{max}} = 80$  for the remaining ones.

We trained the ShapeNet-Chair diffusion prior for 2827 epochs on the ShapeNet-Chair training split with random orthogonal augmentations. During training, we manually adjusted the batch size (ranging from 100 to 200) and learning rate ( $1 \times 10^{-4}$  to  $1 \times 10^{-5}$ ). The ShapeNet-Mixed diffusion prior was also trained on the respective training split with random orthogonal augmentations for 998 epochs with batch size ranging from 200 to 360 and constant learning rate of  $8 \times 10^{-5}$ . For the prior CryoStruct diffusion, we used the training split of Giri et al. (2024) with random rotational augmentations. Throughout the 2070 epochs of training, we manually increased the batch size from 120 to 200 and decreased the learning rate from  $1 \times 10^{-4}$  to  $8 \times 10^{-5}$ .

Table 4: Parameters used during approximate diffusion posterior sampling.

Dataset	Projection points	Number of projections	Coarse grained points	Subunit points	$\beta(t)$	$\alpha$
ShapeNet-Chair	200	5	-	-	$1/t$ if $t > 0.15$ , else 0	10k
ShapeNet-Chair	200	6	-	-	$1/t$ if $t > 0.15$ , else 0	10k
ShapeNet-Mixed	400	5	-	-	$1/t$	10k
ShapeNet-Mixed	400	4	-	-	$1/t$	5k
ShapeNet-Chair	300	1	30	-	$1/t$	4k
ShapeNet-Mixed	300	1	30	-	$1/t$	4k
ShapeNet-Chair	200	2	-	$\approx 256$	$1/t$ if $t > 0.15$ , else 0	4k
ShapeNet-Mixed	200	2	-	$\approx 256$	$1/t$ if $t > 0.15$ , else 0	4k
ShapeNet-Mixed	200	2	30	$\approx 128$	$1/t$ if $t > 0.15$ , else 0	4k
CryoStruct	900	2	30	-	$1/t$	4k
CryoStruct	300	1	40	-	$1/t$	4k
CryoStruct	1024	1	40	-	$1/t$	4k
CryoStruct	1024	3	-	-	$1/t$	40k
CryoStruct	800	4	-	-	$1/t$	40k
CryoStruct	1024	4	-	-	$1/t$	60k
CryoStruct	1024	5	-	-	$1/t$	80k

### A.3 TEST PARAMETERS

We consistently used 40 time steps and  $\rho = 3$  for the reconstruction tasks involving the ShapeNet-Chair, ShapeNet-Mixed, and CryoStruct datasets. An exception was made for the results depicted in Figure 2 where the number of timesteps was doubled to 80. Moreover, we found that our likelihood-based guidance is most effective at lower values of  $t$ . Thus, to prevent superfluous computational work, we have set  $t_{\max}$  to 1 for the reconstruction task. We primarily utilized  $\beta(t) = 1/t$  for most tasks, but noticed that for input projections with a low number of points, setting  $\beta(t)$  to 0 during the final time steps produced sharper 3D point clouds. The guidance strength  $\alpha$  is roughly selected based on the amount of information contained in the input data  $\mathbf{y}$ . Refer to Table 4 for further details.

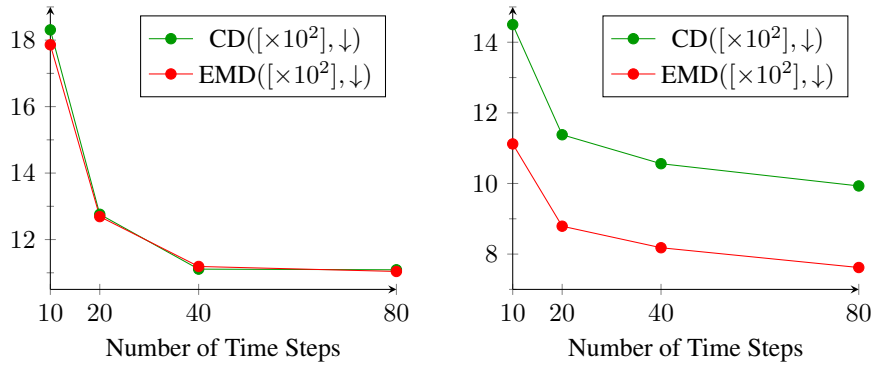
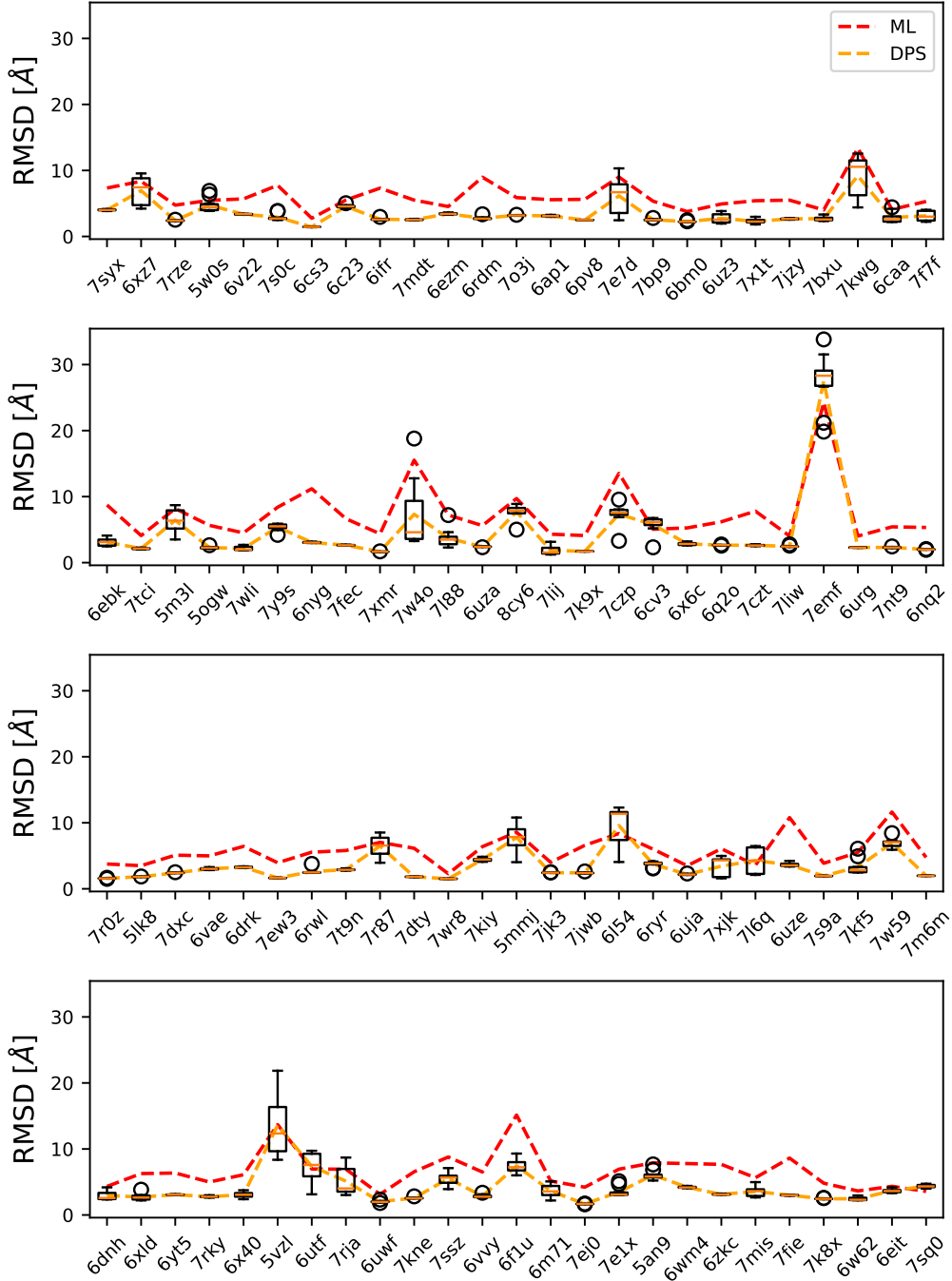


Figure 6: Comparison of reconstruction error in two scenarios over the number of time steps used in Algorithm 1. The plot on the left corresponds to the test case of row 8 in Table 1 with parameters of row 8 in Table 4. The right side depicts the test errors for the configuration outlined in row 3 of Table 1, utilizing the parameters specified in row 4 of Table 4.

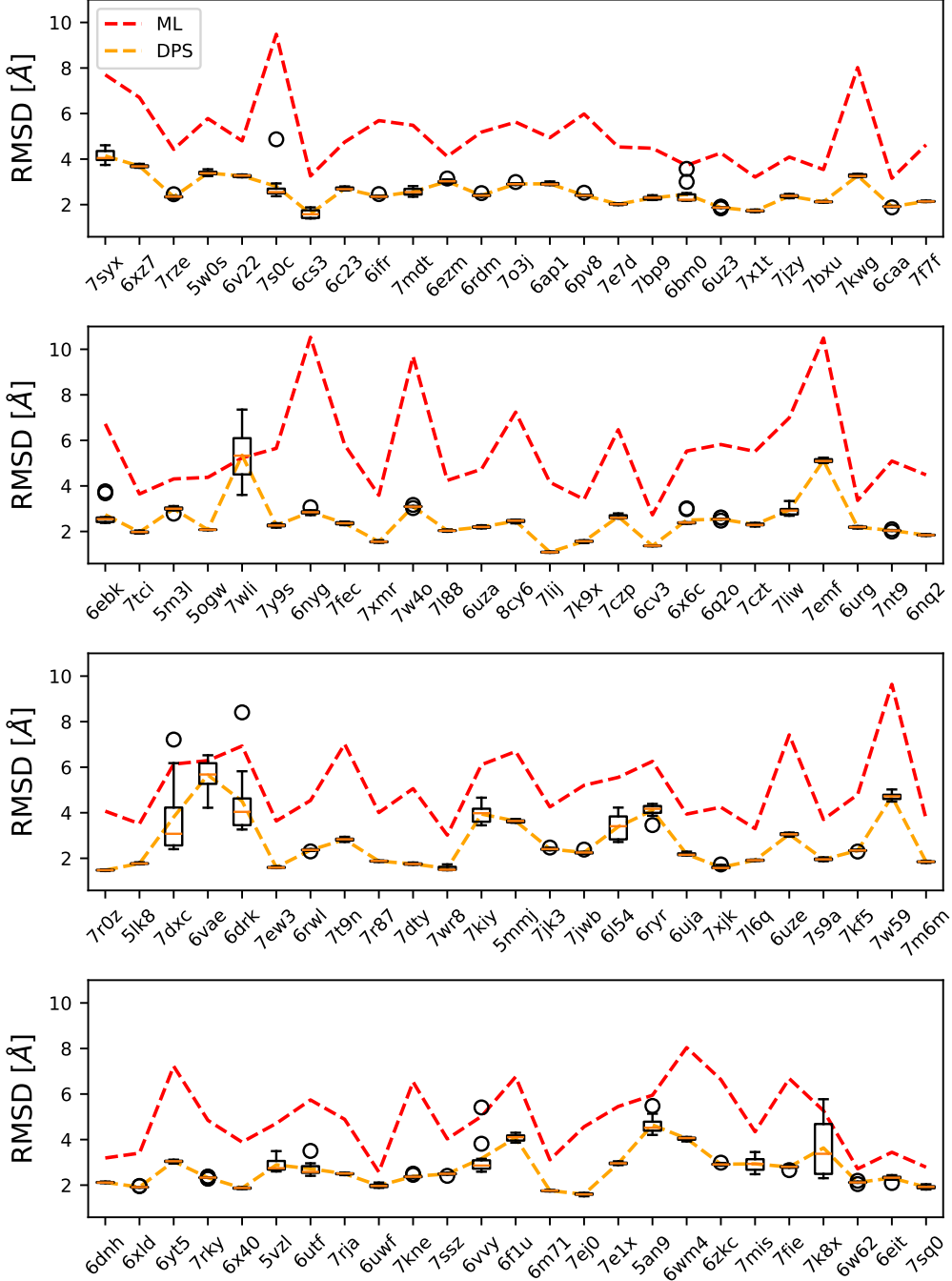
## A.4 CRYOSTRUCT BENCHMARKS

The following figures summarize the CryoStruct benchmark for various input measurements. The PDB codes of the 100 test structures are indicated on the  $x$ -axis. The red dashed line shows the average RMSD of the ML-based models. The box plots and dashed orange lines show the RMSD of the models generated with DPS.

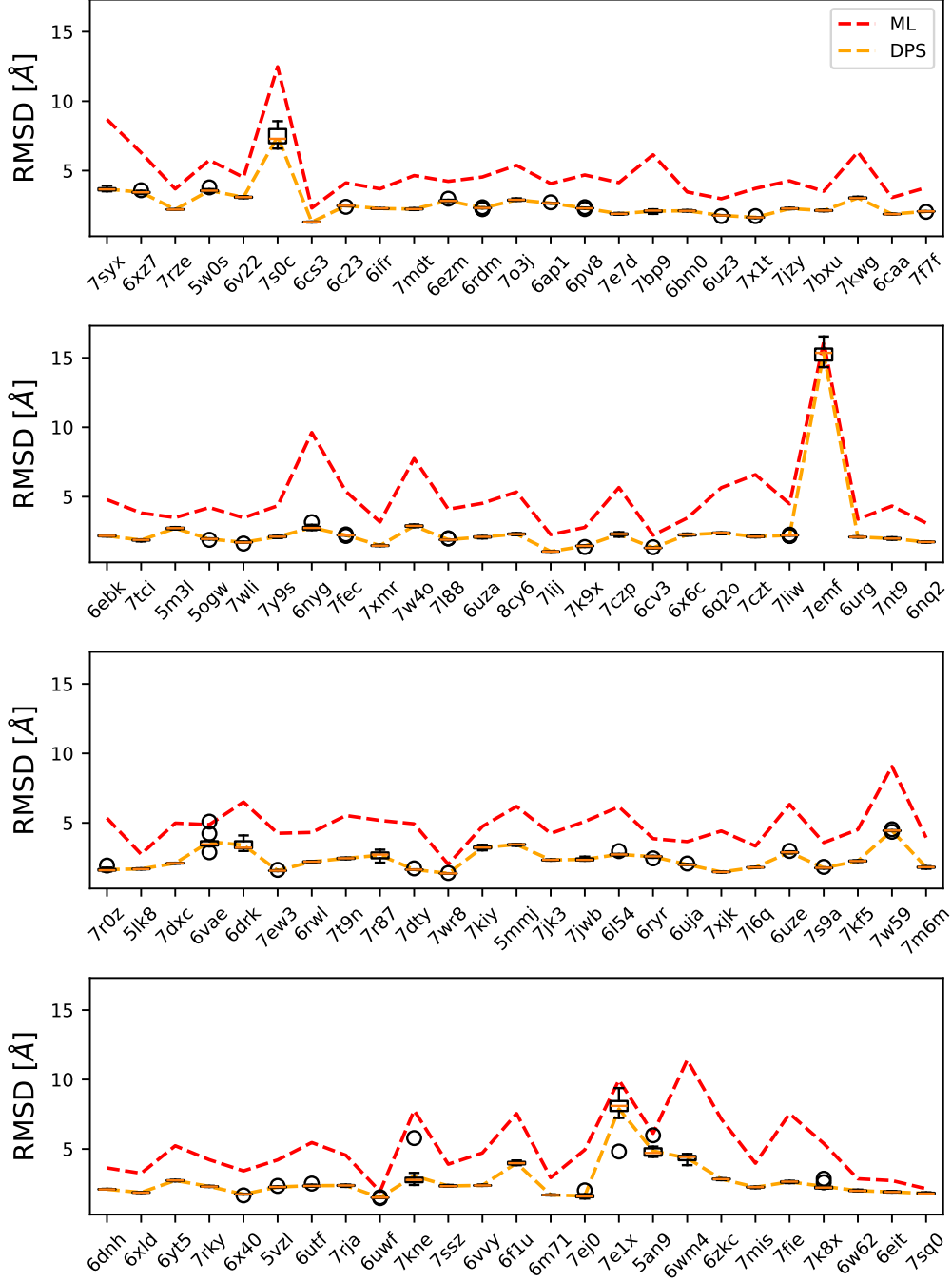
## A.4.1 INPUT DATA: THREE 2D PROJECTIONS, 1024 POINTS PER PROJECTION



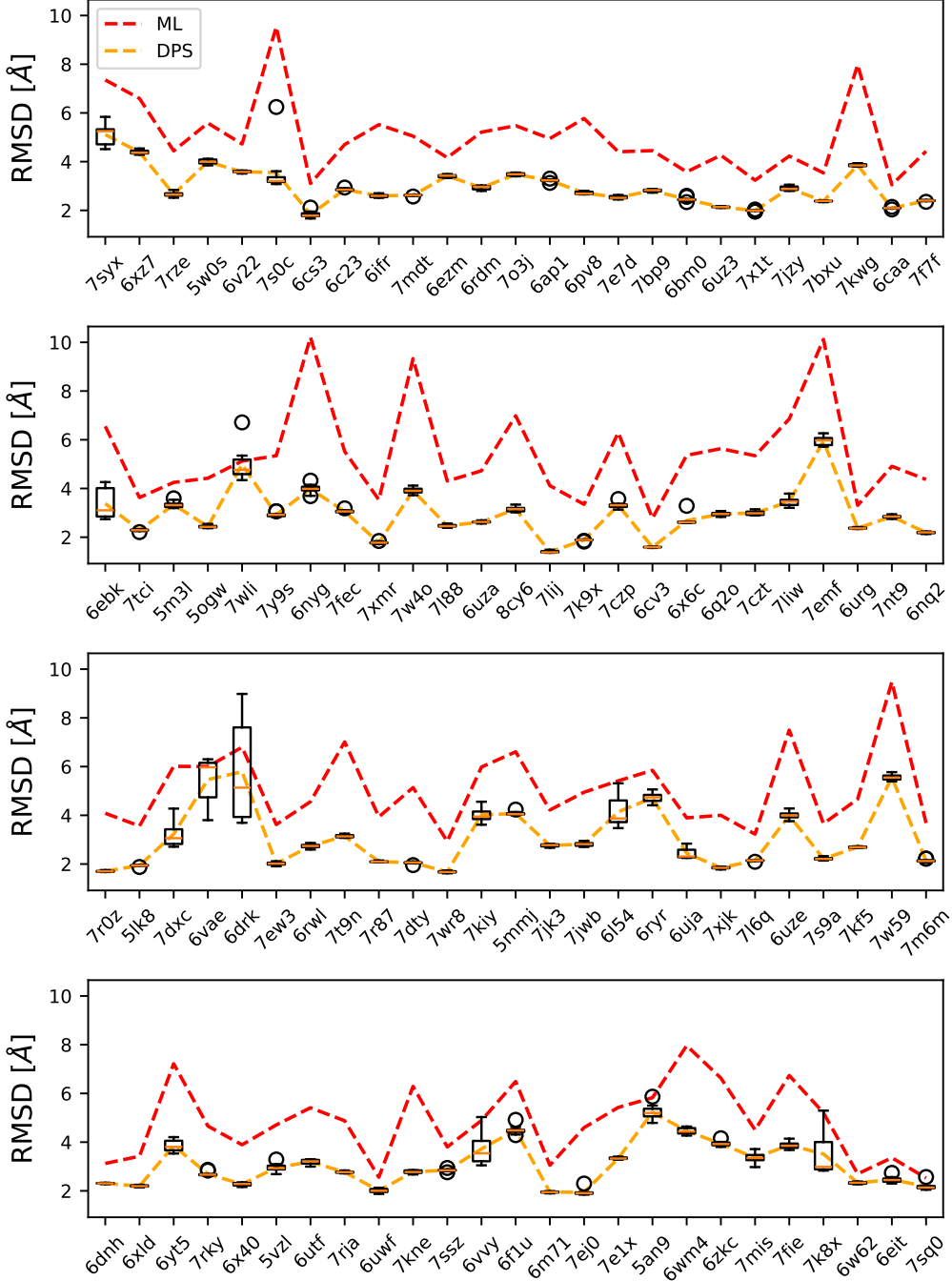
## A.4.2 INPUT DATA: FOUR 2D PROJECTIONS, 1024 POINTS PER PROJECTION



## A.4.3 INPUT DATA: FIVE 2D PROJECTIONS, 1024 POINTS PER PROJECTION

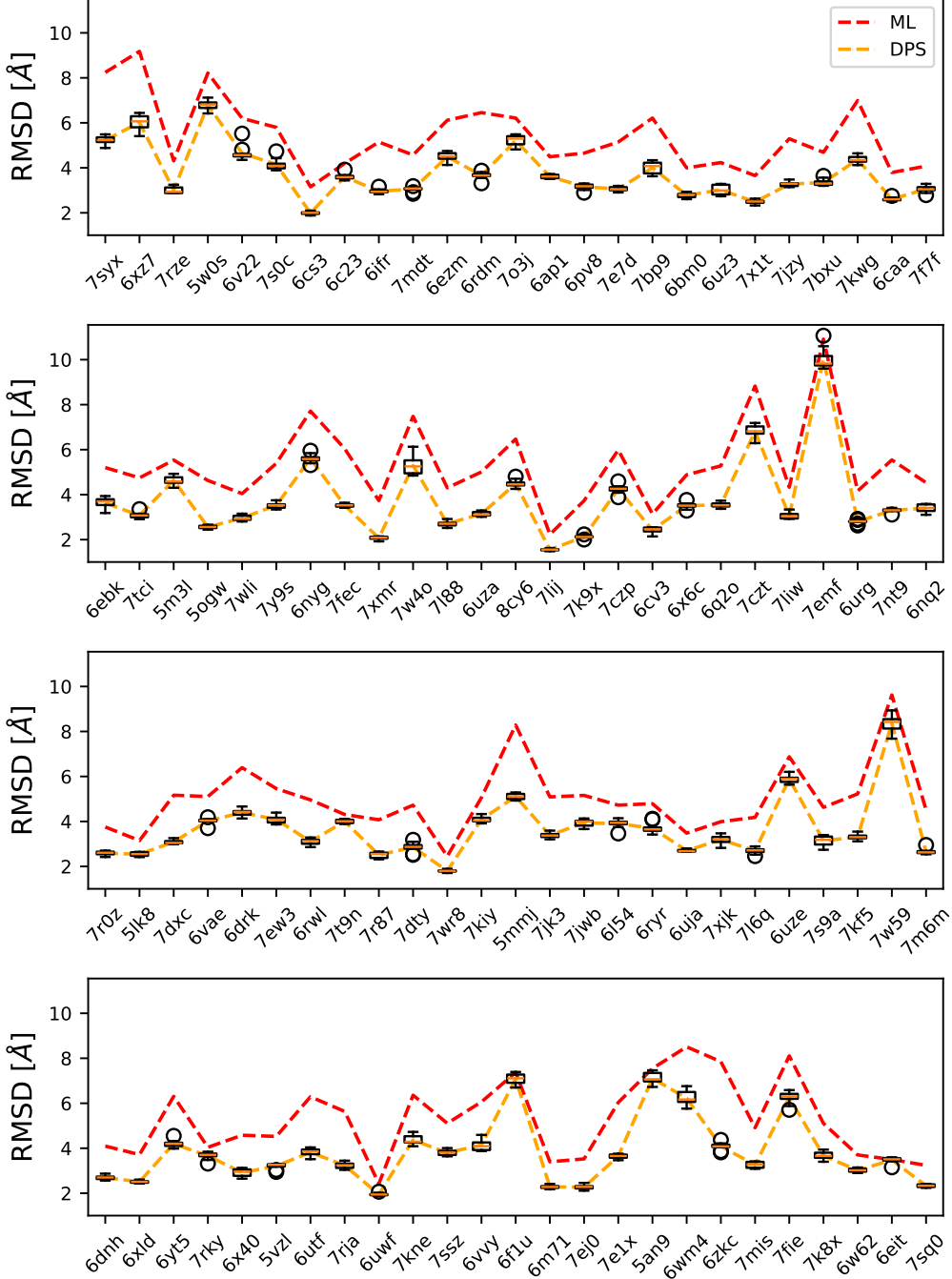


## A.4.4 INPUT DATA: FOUR 2D PROJECTIONS, 800 POINTS PER PROJECTION

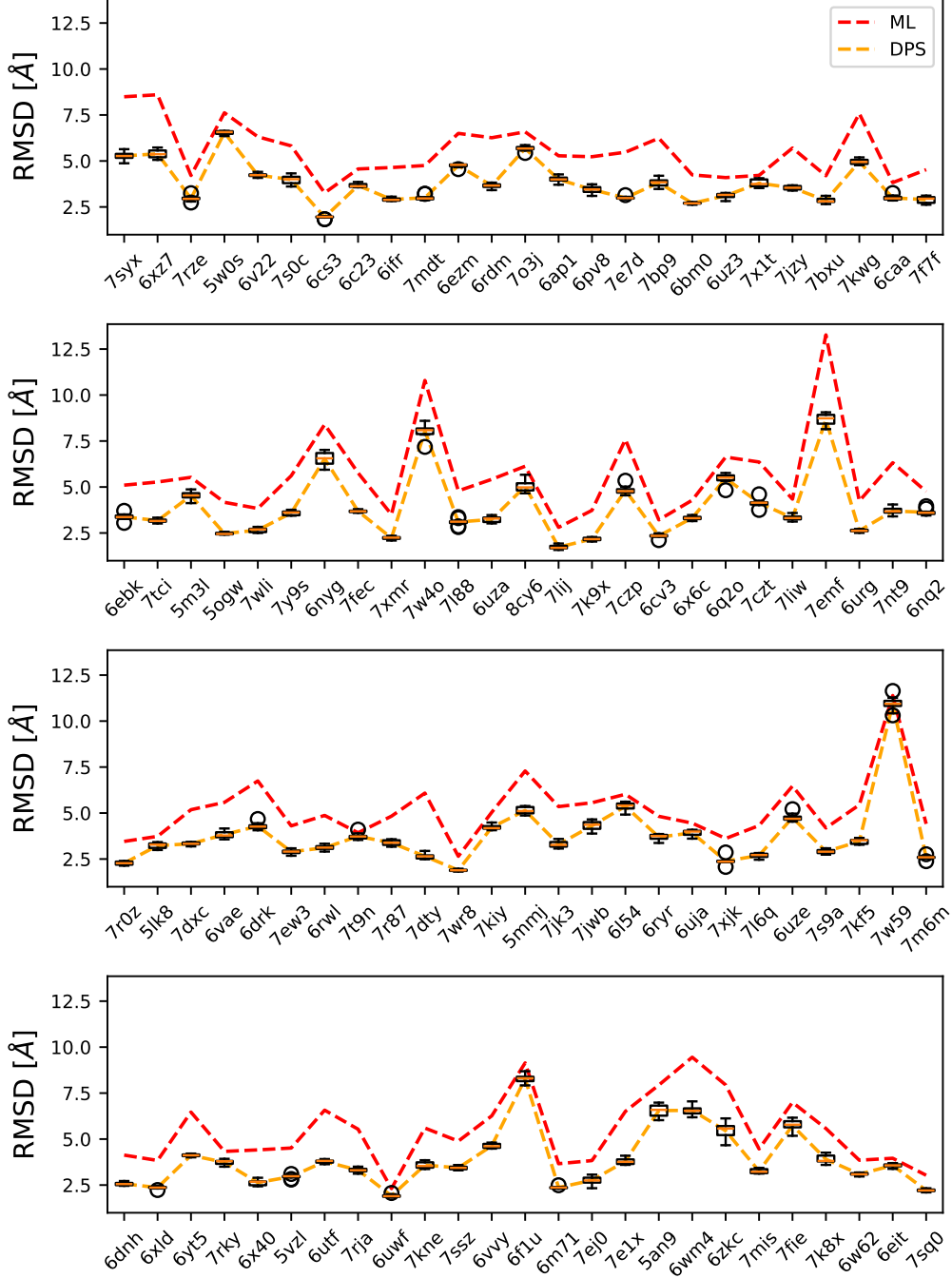




#### A.4.5 INPUT DATA: TWO PROJECTIONS (900 POINTS PER PROJECTION) + A LOW-RESOLUTION STRUCTURE (30 POINTS)



#### A.4.6 INPUT DATA: A SINGLE PROJECTION (1024 POINTS) + A LOW-RESOLUTION STRUCTURE (40 POINTS)



#### A.4.7 INPUT DATA: A SINGLE PROJECTION (300 POINTS) + A LOW-RESOLUTION STRUCTURE (40 POINTS)

