Latent Space Diffusion Models of Cryo-EM Structures

Fan Xiao （樊箫）

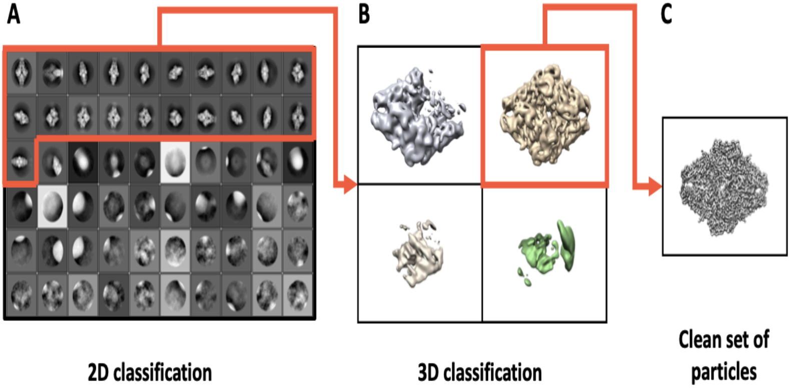
Yau Mathematical Sciences Center,
Tsinghua University

June, 2024

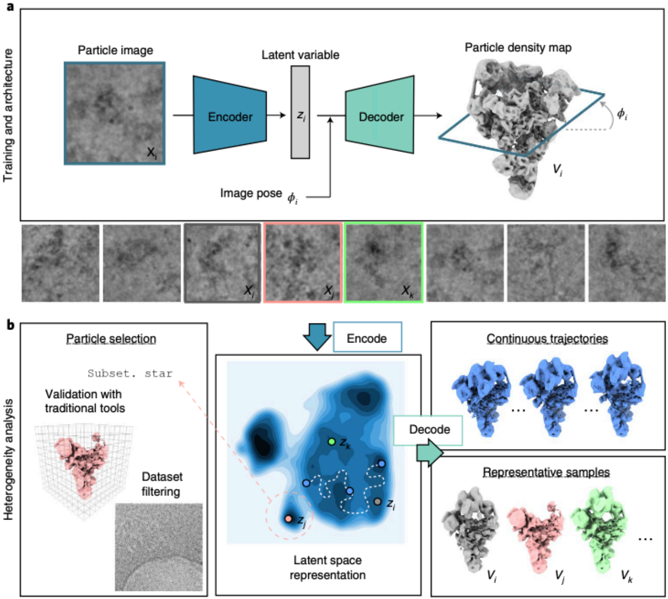
Background of cryo-EM

- Formultion of cryo-EM:

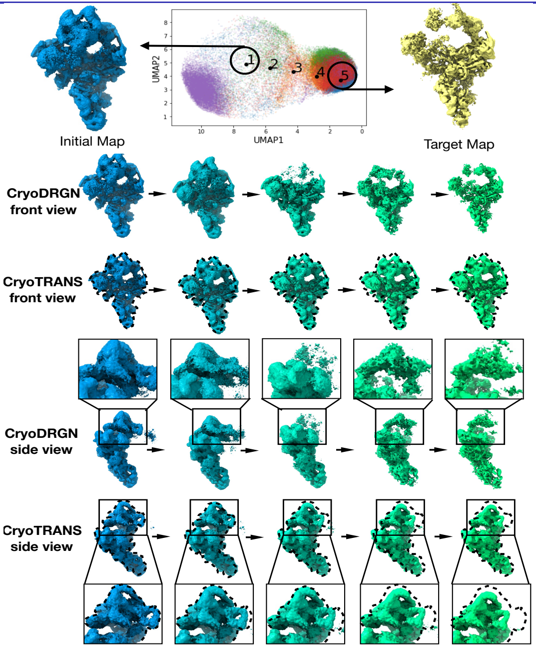
$$X_i = \mathsf{CTF}_i * (P_i^{\phi} \circ V) + N_i \quad i = 1, 2 \ldots N$$



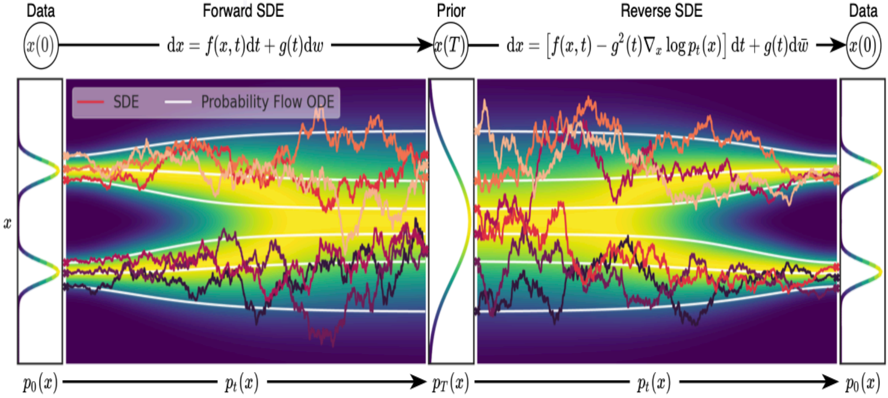
Background of CryoDRGN



Background of CryoDRGN



Background of Diffusion Model



Background of DDPM

- Suppose that $x_0 \sim q_{\text{data}}(x_0)$, the reverse process is defined as a Markov chain:

$$p_\theta\left(\mathbf{x}_{0:T}\right) := p\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$$

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right)$$

- The forward process is fixed to a Markov chain that gradually adds Gaussian noise to the data:

$$q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) := \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$$

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$$

- Define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t := \Pi_{s=1}^{t} \alpha_s$,we have

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right)$$

Background of DDPM

- maximize the ELBO:

$$\mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right)}_{L_T}+$$

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \underbrace{-\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}_{L_0}]$$

- Here

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right)$$

where $\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right):=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t$ and $\tilde{\beta}_t:=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$

- Suppose that $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \sigma_t^2 \mathbf{I}\right)$, we can write

$$L_{t-1}=\mathbb{E}_q\left[\frac{1}{2 \sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)-\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right]+C$$

Background of DDPM

- We choose the paramerization

$$\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right)$$

- The equation smplifies to

$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2 \sigma_t^2 \alpha_t\left(1-\bar{\alpha}_t\right)}\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

- Simplified variational bound:

$$L_{\mathsf{simple}}(\theta):=\mathbb{E}_{t, \mathbf{x}_0, \epsilon}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

Background of DDPM

Algorithm 1 Training

repeat
 $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
 $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 Take gradient descent step on
 $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
 $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
 $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

Background of DDPM

- Define $\beta(\frac{t}{T}) = T\beta_t$, as $T \to \infty$, the forward process of DDPM converges to the following SDE:

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w}$$

- For a general forward SDE:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + \mathbf{G}(t)\mathrm{d}\mathbf{w}$$

there exists a probability flow ODE corresponding to its backward form:

$$\mathrm{d}\mathbf{x} = \left\{\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}\mathbf{G}(t)\mathbf{G}(t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right\} \mathrm{d}t$$

Background of SMLD

- In SMLD, the process $\{\mathbf{x}(t)\}_{t=0}^{1}$ is given by the following SDE:

$$\mathrm{d}\mathbf{x} = \sqrt{\frac{\mathrm{d}\left[\sigma^2(t)\right]}{\mathrm{d}t}}\ \mathrm{d}\mathbf{w}$$

- The probability flow ODE is formulated as:

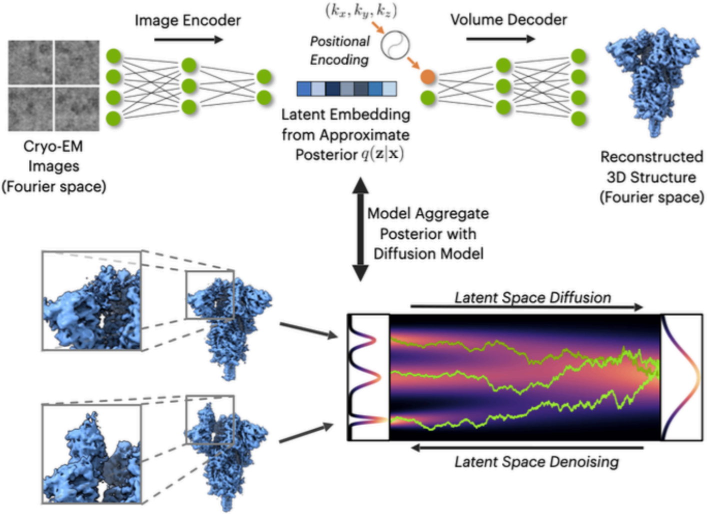
$$d\mathbf{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))dt$$

- Train a denoiser
  $D_\theta(x;\sigma)$($\nabla_{\mathbf{x}} \log p(\mathbf{x};\sigma) \approx s_{\boldsymbol{\theta}} = \left(D_{\boldsymbol{\theta}}(\mathbf{x};\sigma) - \mathbf{x}\right)/\sigma^2$) with:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}\,(\mathbf{x}),\sigma\sim p(\sigma),\mathbf{n}\sim\mathcal{N}(\mathbf{0},\sigma^2)} \left[\lambda(\sigma)\left\|D_{\boldsymbol{\theta}}(\mathbf{x}+\mathbf{n},\sigma) - \mathbf{x}\right\|_2^2\right]$$

Model Outview

- First train a cryoDRGN model, then freeze the parameters and train the diffusion model in latent space



Result

Dataset	D	N	Å/pix.	$\lvert\mathbf{z}\rvert$	Architecture	Epochs
Linear1d	128	50k	6	8	256×3	25
Circular1d	128	100k	6	8	256×3	25
Ribosome (EMPIAR-10076)[42]	256	87,328	1.6375	10	1024×3	50
Covid (Walls et al. [25])	256	276,549	1.640625	8	1024×3	25

Diffusion Model

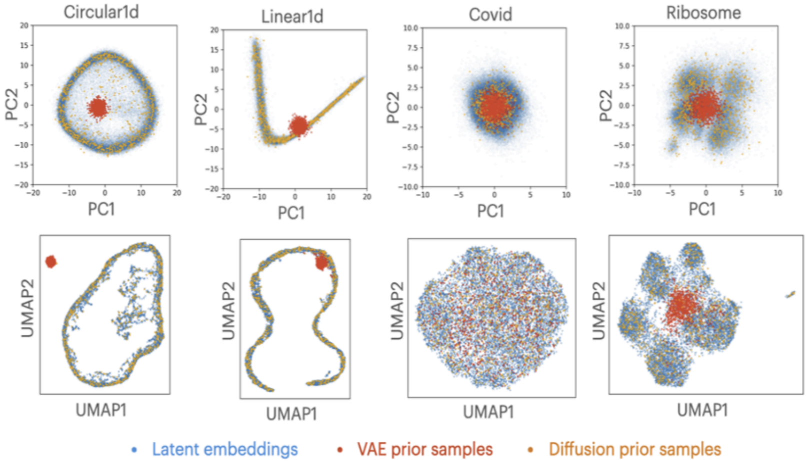
- Network Arhitecture: ResNet Architecture with 16 hidden layers, using 128 hidden dimensions per layer
- Training Algorithm: adam
- Batch size:1024
- Sampling: use an adaptive step-size Runge-Kutta solver for the Probability Flow ODE
- Regularize Sampling:

$$d\mathbf{x} = \underbrace{-\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p(\mathbf{x};\sigma(t))dt}_{\text{Probability Flow ODE; see Eq. (2)}}$$

$$- \underbrace{\beta(t)\sigma^2(t)\nabla_{\mathbf{x}} \log p(\mathbf{x};\sigma(t))dt + \sqrt{2\beta(t)}\sigma(t)d\omega_t}_{\text{Langevin diffusion component}},$$

Diffusion Prior

- Diffusion model prior can model the latent distribution of data



Dataset	Latent Diffusion Model Prior	Standard Gaussian VAE Prior
circular1d	**0.015**	0.82
linear1d	**0.015**	0.86
covid	**0.005**	0.19
ribosome	**0.015**	0.58

Diffusion Prior

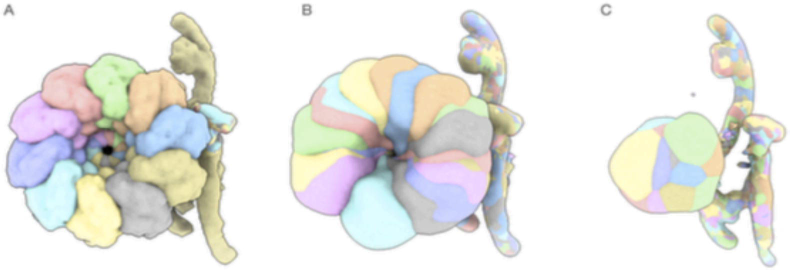
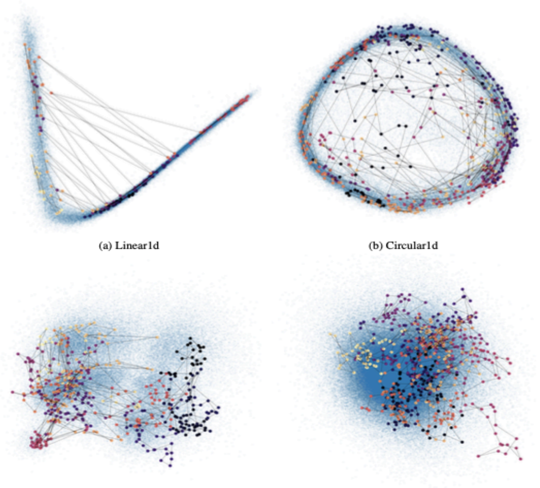


Figure 3: *circular1d* volumes viewed as isosurface contour. **A**: Ground truth volumes. **B**: 20 volumes sampled from diffusion model prior. **C**: 20 volumes sampled from standard Gaussian VAE prior.

Diffusion Prior

- Run Langevin Dynamics in the latent space

$$\hat{\mathbf{z}}_{n+1} = \hat{\mathbf{z}}_n - \frac{\hat{\mathbf{z}}_n}{2}\Delta t + \sqrt{\Delta t}\mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

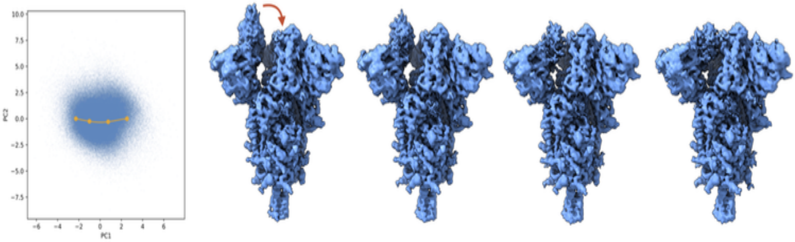


(a) Linear1d (b) Circular1d

(c) Ribosome (d) Covid

Diffusion Inteporlation

- Given z_A and z_B in the latent space, encode $\hat{z}_{AB}(y) = (1-y)\hat{z}_A + y\hat{z}_B$, then decode $\hat{z}_{AB}(y)$ back to $z_{AB}(y)$
- Use the Probability FLow ODE to sample



Diffusion Inteporlation

- Diffusion model has learnt a sharp transition in its Probability Flow ODE

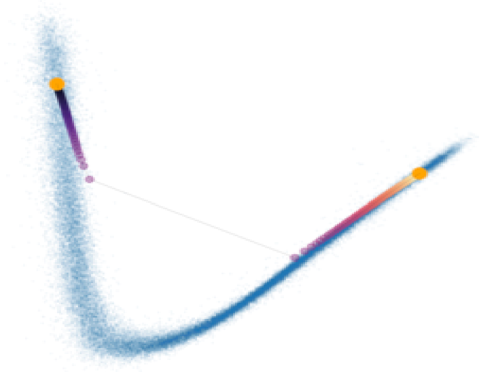


Figure 4: Conformational state interpolation with the diffusion model prior for *linear1d*; Also see Fig. 8.

Diffusion Inteporlation

- Diffusion model has learnt a sharp transition in its Probability Flow ODE

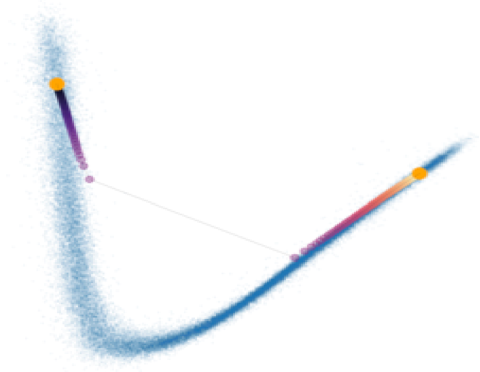


Figure 4: Conformational state interpolation with the diffusion model prior for *linear1d*; Also see Fig. 8.

Future work

- Conditional Generative Modeling
  - Build on classifier and classifier-free guidance technique
- Coupling with Atomic Models
  - Combined with future version of CryoDRGN
- Molecular Simulation
- End-to-end Training and Hierarchical Models
  - End-to-end training of cryoDRGN together with the latent diffusion model
  - Learn one diffusion model to capture the main data clusters and another to capture the intra-cluster variation

Thank you!