3D DDPM

A Preprint

Authors suggest changes to diffusion process, inspired by non-equilibrium thermodynamics.

non-equilibrium thermodynamics(it deals with systems that are actively changing and have energy or matter flowing through them). Authors regard points in point cloud as particles in a non-equilibrium thermodynamic system in contact with a heat bath. Analogously, authors connect original point distribution to a noise distribution using diffuision process(DDPM particulary). oint clouds can be regarded as samples from a point distribution. This viewpoint inspires exploration on applying likelihood-based methods to point cloud modeling and generation.

Formulate the reverse diffusion process for generation as:

$$p_\theta(x_{0:T} \mid z) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t, z), \tag{1}$$

where each transition is modeled as:

$$p_\theta(x_{t-1} \mid x_t, z) = \mathcal{N}(x_{t-1} \mid \mu_\theta(x_t, t, z), \beta_t \mathbf{I}). \tag{2}$$

As training objective the same ELBO as in DDPM is used with the same transformation so i don t find any reason to write about this equation.

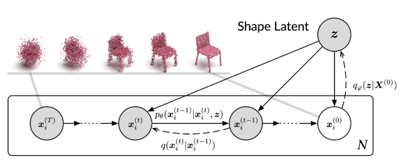

Figure 2. The directed graphical model of the diffusion process for point clouds. N is the number of points in the point cloud $X^{(0)}$.

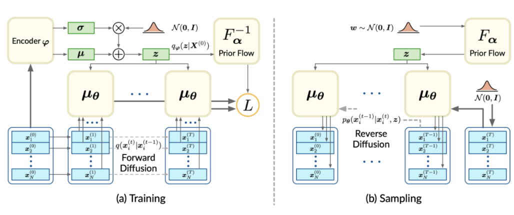

Figure 3. The illustration of the proposed model. (a) illustrates how the objective is computed during the training process. (b) illustrates the generation process.

For point cloud generation authors sugget bijective F_α that maps an isotropic Gaussian to a complex distribution. More specifically, exact probability of the target distribution can be computed by the change-of-variable formula:

$$p(z) = p_w(w) \cdot | \det \frac{\delta F_\alpha}{\delta w} |^{-1}, \quad w = F_\alpha^{-1}(z)$$

For encoder $q_\varphi(z|X^{(0)})$ Pointnet is used.

Algorithm 1 Correct 3D Diffusion Generation

Require: Diffusion model p_θ, prior network F_α, number of steps T, number of points N, variance schedule $\{\beta_t\}_{t=1}^{T}$

```
                                        ▷ Step 1: Create a shape blueprint
Sample w ~ N(0, I)
Compute latent code z = F_α(w)
                                        ▷ Step 2: Initialize with pure noise at time T
Sample X^(T) ~ N(0, I)                  ▷ THIS IS THE CORRECT INITIALIZATION
                                        ▷ Step 3: Run the reverse process to denoise
for t = T down to 1 do
    Predict mean μ_θ = p_θ(X^(t), t, z)
    Sample X^(t−1) ~ N(μ_θ, β_t I)
end for
                                        ▷ Step 4: Return the final result
return Generated point cloud X^(0)
```