DCE-diff: Diffusion Model for Synthesis of Early and Late Dynamic Contrast-Enhanced MR Images from Non-Contrast Multimodal Inputs

Supplementary Material

7. Appendix

From [31], we adapt the diffusion model training and inference.

The forward diffusion process is a Markovian process that adds noise to the image $\boldsymbol{y}_0 \equiv \boldsymbol{y}$ over T iterations. At a time step t, the addition of noise is given by:

$$q\left(\boldsymbol{y}_{t+1} \mid \boldsymbol{y}_t\right) = \mathcal{N}\left(\boldsymbol{y}_{t-1}; \sqrt{\alpha_t}\boldsymbol{y}_{t-1}, (1-\alpha_t) I\right) \tag{3}$$

$$q\left(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0\right) = \prod_{t=1}^{T} q\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}\right) \tag{4}$$

where α_t are noise schedule hyper-parameters. At $t = T, \boldsymbol{y}_T$ is Gaussian Noise. The forward process can be marginalizable at each step and is given by

$$q\left(\boldsymbol{y}_t \mid \boldsymbol{y}_0\right) = \mathcal{N}\left(\boldsymbol{y}_t; \sqrt{\gamma_t}\boldsymbol{y}_0, (1-\gamma_t) I\right) \tag{5}$$

where $\gamma_t = \prod_{t'}^{t} \alpha'_t$.

$$q\left(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t\right) = \mathcal{N}\left(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) \tag{6}$$

where $\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t}\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t}\boldsymbol{y}_t$ and $\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}$.

During reverse Process:

$$\widetilde{\boldsymbol{y}} = \sqrt{\gamma}\boldsymbol{y}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{7}$$

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\mathbb{E}_{\boldsymbol{\epsilon},\gamma}\| f_\theta(\boldsymbol{x}, \underbrace{\sqrt{\gamma}\boldsymbol{y}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}}_{\widetilde{\boldsymbol{y}}}, \gamma) - \boldsymbol{\epsilon}\|_p^p \tag{8}$$

Inference: The model performs inference via the learned reverse process. Since the forward process is constructed so the prior distribution $\mathrm{p}\left(y_T\right)$ approximates a standard normal distribution $\mathcal{N}\left(\boldsymbol{y}_T \mid \boldsymbol{0}, \boldsymbol{I}\right)$, the sampling process can start at pure Gaussian noise, followed by T steps of iterative refinement.

The neural network model f_θ is trained to estimate $\boldsymbol{\epsilon}$, given any noisy image $\widetilde{\boldsymbol{y}}$, and $\boldsymbol{y}_t$. Thus, given $\boldsymbol{y}_t$, we approximate $\boldsymbol{y}_0$ as

$$\hat{\boldsymbol{y}}_0 = \frac{1}{\sqrt{\gamma_t}}\left(\boldsymbol{y}_t - \sqrt{1-\gamma_t} f_\theta\left(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t\right)\right) \tag{9}$$

Substitute the estimate $\hat{\boldsymbol{y}}_0$ into the posterior distribution of $q\left(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t\right)$ to parameterize the mean of $p_\theta\left(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x}\right)$ as

Algorithm 1 Training a denoising model f_θ

repeat
$(\boldsymbol{x}, \boldsymbol{y}_0) \sim p(\boldsymbol{x}, \boldsymbol{y})$
$\gamma \sim p(\gamma)$
$\epsilon \sim \mathcal{N}(0, \mathrm{I})$
Take a gradient descent step on
$\nabla_\theta \left\| f_\theta\left(\boldsymbol{x}, \sqrt{\gamma}\boldsymbol{y}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}, \gamma\right) - \boldsymbol{\epsilon}\right\|_p^p$
until converged

Algorithm 2 Inference in T iterative refinement steps

$y_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots, 1$ do
$z \sim \mathcal{N}(0, \mathrm{I})$ if $t > 1$, else $z = 0$
$\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta\left(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t\right)\right) + \sqrt{1-\alpha_t}\boldsymbol{z}$
end for
return y_0

$$\mu_\theta\left(x, \boldsymbol{y}_t, \gamma_t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta\left(x, \boldsymbol{y}_t, \gamma_t\right)\right) \tag{10}$$

The variance $p_\theta\left(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x}\right)$ is set to $(1-\alpha_t)$, a default. Now, each iteration of the reverse process can be written as

$$\boldsymbol{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta\left(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t\right)\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \tag{11}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$. This resembles one step of Langevin dynamics for which f_θ provides an estimate of the gradient of the data log density.