ENHANCING PROBABILISTIC DIFFUSION MODELS WITH VARIATIONAL INFERENCE

BY RAVI RAJ KUMAR

OUTLINE

INTRODUCTION

- **Diffusion-based generative models**
  - Corrupt data by gradually adding Gaussian noise
  - Learn to reverse that process to synthesize images
- **State-of-the-art sample fidelity**
  - Photorealistic generation in images, audio, video
- **BUT… limited likelihood performance**
  - Struggle to match autoregressive models on bits-per-dimension
- **Our goal**
  - Bring diffusion models up to par on likelihood
  - Retain their sample quality

WHY LIKELIHOOD MATTERS

- **Beyond visual quality**
  - Compression: lower BPD $\Rightarrow$ better lossless coding
  - Density estimation: scientific modeling, anomaly detection
- **Bits-per-dimension**
  - Unified metric for sample fidelity *and* statistical modeling
- **Bridging the gap**
  - Retain diffusion's generative power
  - Achieve AR-level likelihood

THREE VIEWS OF DIFFUSION PROCESS

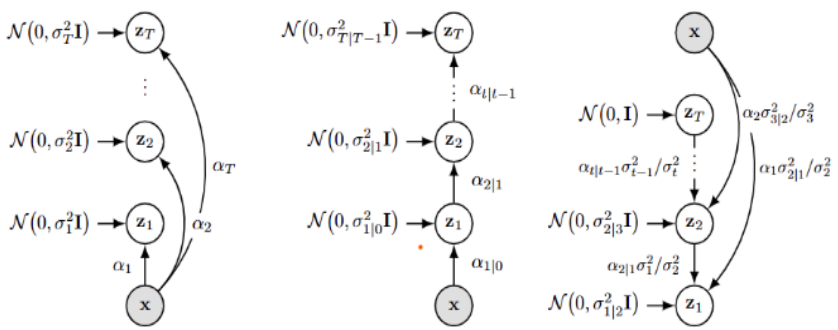
- 1. Forward Parameterization

$$q(\mathbf{z}_t \mid \mathbf{x}) = \mathcal{N}\big(\mathbf{z}_t;\, \alpha_t\, \mathbf{x},\, \sigma_t^2\, \mathbf{I}\big),$$

- 2. Markov Chain

$$q(\mathbf{z}_t \mid \mathbf{z}_s) = \mathcal{N}\big(\mathbf{z}_t;\, \alpha_{t|s}\, \mathbf{z}_s,\, \sigma_{t|s}^2\, \mathbf{I}\big)$$

- 3. Top-Down Posterior



(a) Gaussian Diffusion (b) Markovian Transitions (c) Top-down Posterior

$$q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}) = \mathcal{N}\big(\mathbf{z}_s;\, \mu_Q(\mathbf{z}_t, \mathbf{x}; s, t),\, \sigma_Q^2(s,t)\, \mathbf{I}\big).$$

THREE VIEWS OF DIFFUSION PROCESS

Parameter	Expression
$\alpha_{t\|s}$	α_t / α_s
$\sigma^2_{t\|s}$	$\sigma_t^2 - \alpha^2_{t\|s}\sigma_s^2$
$\mu_Q(\mathbf{z}_t, \mathbf{x}; s, t)$	$\frac{\alpha_{t\|s}\sigma_s^2}{\sigma_t^2}\mathbf{z}_t + \frac{\alpha_s\sigma^2_{t\|s}}{\sigma_t^2}\mathbf{x}$
$\sigma_Q^2(s,t)$	$\frac{\sigma^2_{t\|s}\sigma_s^2}{\alpha^2_{t\|s}\sigma_s^2 + \sigma^2_{t\|s}}$

Leverage all three diffusion perspectives (Forward, Markov-Chain, Top-Down Posterior)

- **Transition Scale**
- **Transition Variance**
- **Posterior Mean**
- **Posterior Variance**

- Exact Gaussian parameters for each reverse step

- enabling efficient sampling and likelihood evaluation

NOISE SCHEDULE

- **Purpose:** Controls how much noise is added at each diffusion timestep
- **Fixed vs. Learnable:**
- Fixed schedules are hand-tuned (e.g. linear, cosine)
- Learnable schedules adapt to data for variance reduction
- **Variance Preservation:** Ensures latent statistics match input data
- Schedule defines a signal-to-noise ratio curve
- **Continuous-Time Insight:** VLB invariant to schedule except at endpoints

$$\sigma_t^2 = \mathrm{sigmoid}(\gamma_\eta(t)),$$

$$\mathrm{SNR}(t) = \frac{\alpha_t^2}{\sigma_t^2} = \exp(-\gamma_\eta(t)).$$

- **Linear Schedule:**
  - $\gamma(t) = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})\, t$
  - Simple, easy to implement and tune

REVERSE PROCESS & TRAINING OBJECTIVE

- Training Objective (VLB)

$$-\log p(x) \;\le\; \underbrace{D_{\mathrm{KL}}\big(q(z_T \mid x) \,\|\, p(z_T)\big)}_{\text{Prior Term}} \;+\; \underbrace{\mathbb{E}_{q(z_0|x)}\big[-\log p(x \mid z_0)\big]}_{\text{Reconstruction Term}} \;+\; \underbrace{L_T(x)}_{\text{Diffusion Term}} \;.$$

$$L_T(x) = \sum_{i=1}^{T} \mathbb{E}_{q\left(z_{t(i)}|x\right)} \Big[D_{\mathrm{KL}}\big(q\big(z_{s(i)} \mid z_{t(i)}, x\big) \,\|\, p\big(z_{s(i)} \mid z_{t(i)}\big)\big) \Big].$$

- Reverse Transition

$$p(\mathbf{z}_s \mid \mathbf{z}_t) = \mathcal{N}\big(\mathbf{z}_s;\, \mu_\theta(\mathbf{z}_t; s, t),\, \sigma_Q^2(s,t)\, I\big).$$

$$\mu_\theta(\mathbf{z}_t; s, t) = \frac{\alpha_{t|s}\, \sigma_s^2}{\sigma_t^2}\, \mathbf{z}_t + \frac{\alpha_s\, \sigma_{t|s}^2}{\sigma_t^2}\, \hat{x}_\theta(\mathbf{z}_t; t),$$

DISCRETE VS CONTINUOUS MODEL

- Discrete-time loss

$$D_{\mathrm{KL}}(q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}) \parallel p(\mathbf{z}_s \mid \mathbf{z}_t)) = \frac{1}{2}\left(\mathrm{SNR}(s) - \mathrm{SNR}(t)\right) \|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2 .$$

$$\mathcal{L}_T(\mathbf{x}) = \frac{T}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}), i\sim U\{1,T\}}\left[\mathrm{expm1}\left(\gamma_{\boldsymbol{\eta}}(t) - \gamma_{\boldsymbol{\eta}}(s)\right) \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}; t)\|_2^2\right] .$$

- Continuous -time loss

$$L_\infty(x) = -\frac{1}{2}\,\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I),\, t\sim\mathrm{Uniform}(0,1)}\left[\mathrm{SNR}'(t)\,\frac{\|x - \hat{x}_\theta(z_t; t)\|_2^2}{2}\right] .$$

$$L_\infty(x) = \frac{1}{2}\,\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I),\, t\sim\mathrm{Uniform}(0,1)}\left[\gamma'_\eta(t)\,\frac{\|\varepsilon - \hat{\varepsilon}_\theta(z_t; t)\|_2^2}{2}\right] ,$$

$$\gamma'_\eta(t) = \frac{d\,\gamma_\eta(t)}{dt} .$$

DATASET USED

Datasets:-

- MNIST
- FashionMNIST

Preprocessing

- Pixel values scaled to [0,1]
- Zero-mean, unit-variance normalization for variance preservation

Data Splits & Augmentation

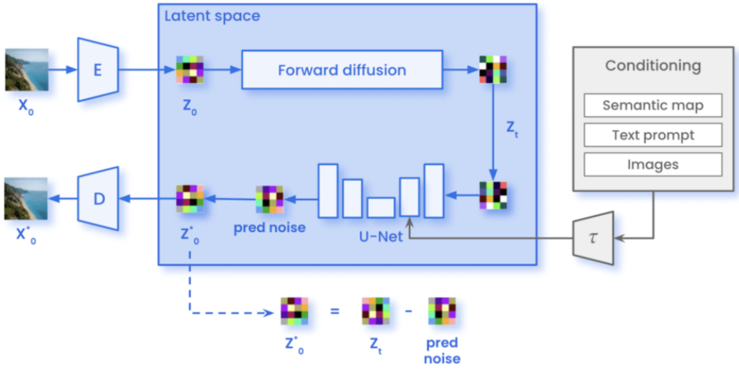
- Held-out validation set (e.g. 10% of training data) for hyperparameter tuning
- Optional augmentations: small rotations, translations, flips to boost robustness

MODEL ARCHITECTURE

.Key components implemented:

- **Encoder:** Maps input images to a latent space.
- **Decoder:** Reconstructs images from the latent representation.
- **ScoreNet:** Predicts noise during the reverse process.

- Employed a linear noise schedule to validate model functionality



TRAINING

Trained the model using the AdamW optimizer for 20,000 steps in PyTorch.

Training

- Fetch a batch of images
- Add noise based on timesteps
- Predict noise and compute error
- Update model via backpropagation

Sampling

- Initialize with random noise
- Iteratively denoise using ScoreNet
- Return the final image

Algorithm 1 VDM Training (Concise)

repeat
    Sample $x_0 \sim q(x)$, $t \sim \mathcal{U}[0,1]$, $\varepsilon \sim \mathcal{N}(0, I)$
    Compute $\gamma_t, \quad \alpha_t = \sqrt{\text{sigmoid}(-\gamma_t)}, \quad \sigma_t = \sqrt{\text{sigmoid}(\gamma_t)}$
    Form $z_t = \alpha_t x_0 + \sigma_t \varepsilon$ and predict $\hat{\varepsilon} = \varepsilon_\theta(z_t, t)$
    Loss $L = \frac{1}{2} \gamma'_t \|\varepsilon - \hat{\varepsilon}\|^2$, update $\theta \leftarrow \theta - \eta \nabla_\theta L$
until max steps

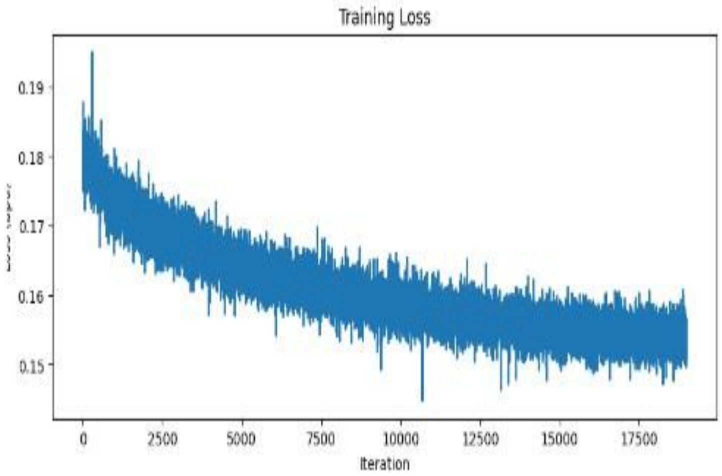
Algorithm 2 VDM Sampling (Concise)

Initialize $z_T \sim \mathcal{N}(0, I)$
for $i = T, \ldots, 1$ **do**
    Set $t_i = i/T$, compute $\gamma_i,\ \alpha_i,\ \sigma_i$; predict $\hat{\varepsilon} = \varepsilon_\theta(z_i, t_i)$
    Estimate $\hat{x}_0 = (z_i - \sigma_i \hat{\varepsilon})/\alpha_i$
    $z_{i-1} = \alpha_{i-1}\left(\frac{\alpha_i^2 - \sigma_i^2}{\alpha_i^2}\, z_i + \frac{\sigma_i^2}{\alpha_i^2}\, \hat{x}_0\right) + \sigma_{i-1}\sqrt{1 - \frac{\alpha_{i-1}^2}{\alpha_i^2}}\, \xi$
end for
return $\hat{x}_0$

RESULTS

- The model shows a steady reduction in the normalized loss in BPD over 20K iterations.
- Early losses are high and gradually decrease to around 0.25 bpd.

.



RESULTS

- VAE - MNIST - ~0.98
- PixelCNN - MNIST - ~1.00
- DDPM - MNIST - ~0.95
- This Model - MNIST/FashionMNIST - 0.31

Model	Dataset	BPD (bits/dim)	Reference
Variational Autoencoder (VAE)	MNIST	~0.98	Kingma & Welling (2013)
PixelCNN (Autoregressive)	MNIST	~1.00	van den Oord et al. (2016)
DDPM (Diffusion model)	MNIST	~0.95	Ho et al. (2020)
Our Discrete Diffusion Model (Diffusion model)	MNIST	**0.25**	This work
Variational Autoencoder (VAE)	FashionMNIST	~1.20	Xiao et al. (2017)
PixelCNN (Autoregressive)	FashionMNIST	~1.30	Various benchmarks
Our Discrete Diffusion Model (Diffusion model)	FashionMNIST	**0.31**	This work

GENERATED IMAGES

FASHIONMNIST

MNIST

CONCLUSION

- Our experiments show that the model achieves effective training with normalized loss values around 0.25 bpd.
- The generated images confirm the model's ability to reverse the diffusion process and reconstruct high-quality images.
- Future research will aim to refine the noise schedule further and extend the model's applicability.


THANK YOU