latent space 에서의 SGM.

1. 새로운 score-matching objective.

2. 새로운 parameterization

3. 새로운 technic

1. Introduction

likelihood의 경우 data 분포를 잘 학습하면서도, high-fidelity 인 data를 생성하는 것이 고려되었다.

SGM에서 data의 score 추정이 고려되었고, NCSN++에서 SDE가 maximum likelihood로 바꿔서리 설명했다.

SGM은 cost가 많이 드는데, 간단한 noise 분포에서, 복잡한 분포로, SDE 나 probability flow ODE로 풀기 때문이다.

미분 방정식은 복잡하고, 정확하게 풀려면, step를 작게 나눠야하기 때문에 수천개의 neural evaluate가 필요하다.

generation complexity는 data distribution과 forward SDE에 의해 결정되며, → 단계가 적으면 생능

SDE-based model은 continuous이기 때문에 binary나 categorical, graph-structure data에 따라 적용하기 힘들다.

LSGM은 VAE기반 latent score-based를 제안한다.

embedding space 생성 후 SGM은 embedding 분포학습

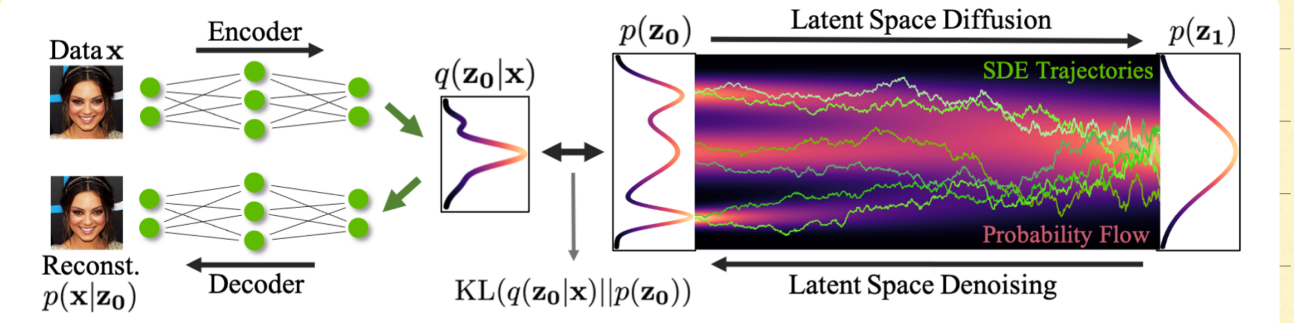


Figure 1: In our latent score-based generative model (LSGM), data is mapped to latent space via an encoder $q(\mathbf{z}_0|\mathbf{x})$ and a diffusion process is applied in the latent space ($\mathbf{z}_0 \rightarrow \mathbf{z}_1$). Synthesis starts from the base distribution $p(\mathbf{z}_1)$ and generates samples in latent space via denoising ($\mathbf{z}_0 \leftarrow \mathbf{z}_1$). Then, the samples are mapped from latent to data space using a decoder $p(\mathbf{x}|\mathbf{z}_0)$. The model is trained end-to-end.

* Synthesis Speed.

VAE를 우선 gaussian에 대한 prior로 pre-train 시킨다.

└ prior가 diffusion의 noise의 비슷하도록 ⇒ diff가 쉬워짐.

* Expressivity

SGM은 neural ODE를 생각할 수 있다.

SGM을 latent의 framework로 생각할 수 있

* Tailored Encoders and Decoders

latent space에서 SGM을 사용하기 때문에 enc, dec의 representation을 증가시킬 수 있다.

LSGM은 ELBO로 훈련 가능

일반 score-matching과 비교 DSM과 target distribution이 동시에 학습되기 때문에 무거나 다 어렵다

기술적 기여 (contribution)

1. VAE와 LSGM을 동시에 학습시키는 DSM objective를 제시.

2. normal distribution과 learnable SGM의 첫인 latent space function의 새로운 parameterization을

도입해서, SGM의 latent의 분포와 normal prior 사이의 불일치만 modeling 할 수 있게 한다.

3. 새로운 SDE 설계와 importance sampling scheme을 도출함으로서, variance reduction tech로

안정적으로 훈련할 수 있도록 한다.

* Background.

NCSN 버전 아닌.

z_0를 start data, z_t를 time t noised perturbed data, $t \in [0,1]$인 continuous라고 할 때.

forward diffusion process $\{z_t\}_{t=0}^{t=1}$ 을 고려

$$\min_{\theta} \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[\lambda(t)\mathbb{E}_{q(\mathbf{z}_0)}\mathbb{E}_{q(\mathbf{z}_t|\mathbf{z}_0)}[||\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t) - \nabla_{\mathbf{z}_t}\log p_\theta(\mathbf{z}_t)||_2^2]\right] \quad (2)$$ → SM

$$\min_{\theta} \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[\lambda(t)\mathbb{E}_{q(\mathbf{z}_0)}\mathbb{E}_{q(\mathbf{z}_t|\mathbf{z}_0)}[||\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t|\mathbf{z}_0) - \nabla_{\mathbf{z}_t}\log p_\theta(\mathbf{z}_t)||_2^2]\right] + C \quad (3)$$ → DSM

$$\mathrm{KL}\big(q(\mathbf{z}_0)||p_\theta(\mathbf{z}_0)\big) \leq \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[\frac{g(t)^2}{2}\mathbb{E}_{q(\mathbf{z}_0)}\mathbb{E}_{q(\mathbf{z}_t|\mathbf{z}_0)}\left[||\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t) - \nabla_{\mathbf{z}_t}\log p_\theta(\mathbf{z}_t)||_2^2\right]\right] \quad (4)$$

↳ DSM을 [11]에 대한 변형.

* Score-based Generative modeling in Lantent space.

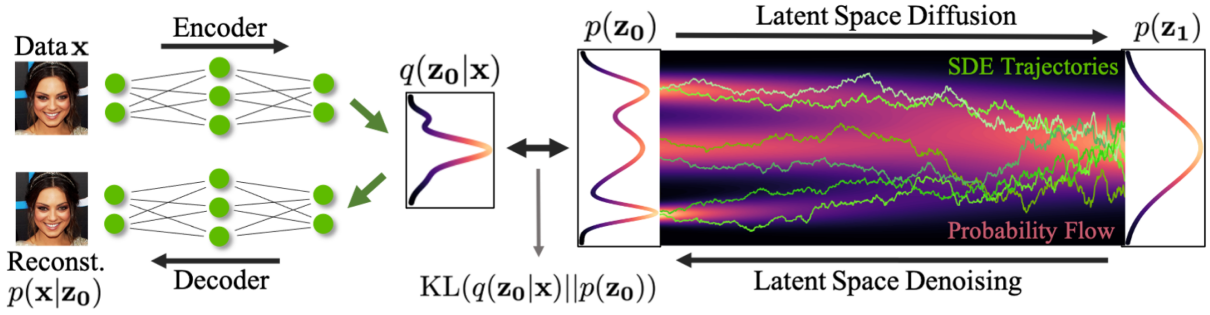


Figure 1: In our latent score-based generative model (LSGM), data is mapped to latent space via an encoder $q(\mathbf{z}_0|\mathbf{x})$ and a diffusion process is applied in the latent space ($\mathbf{z}_0 \rightarrow \mathbf{z}_1$). Synthesis starts from the base distribution $p(\mathbf{z}_1)$ and generates samples in latent space via denoising ($\mathbf{z}_0 \leftarrow \mathbf{z}_1$). Then, the samples are mapped from latent to data space using a decoder $p(\mathbf{x}|\mathbf{z}_0)$. The model is trained end-to-end.

Encoder: ϕ

Diffusion: θ $[0,1]$, $z_1 \sim N(z_1; 0, I)$

Decoder: ψ

Generate : sampling 된 z_0 ($\nabla_{z_t}\log p_\theta(z_t)$로부터) 만 decoder ψ를 이용.

$$P(z_0, x) = p_\theta(z_0)p_\psi(x|z_0)$$

훈련은 NLL의 ELBO로.

$$\mathcal{L}(\mathbf{x}, \phi, \theta, \psi) = \mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[-\log p_\psi(\mathbf{x}|\mathbf{z}_0)\right] + \mathrm{KL}\left(q_\phi(\mathbf{z}_0|\mathbf{x})||p_\theta(\mathbf{z}_0)\right) \tag{5}$$

$$= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[-\log p_\psi(\mathbf{x}|\mathbf{z}_0)\right]}_{\text{reconstruction term}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[\log q_\phi(\mathbf{z}_0|\mathbf{x})\right]}_{\text{negative encoder entropy}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[-\log p_\theta(\mathbf{z}_0)\right]}_{\text{cross entropy}} \tag{6}$$

어려운 점은 SGM의 CE 텀이다.

- The Cross-Entropy Term.

식(5)와 식(6)를 바로 KL이 대입시키지 않는 이유는 바로 적용이 불가능하기 때문이다.

$\nabla_{z_t} \log q(z_t)$가 normalizing flow 같은 일반적인 non-normal distribution에 대해 분석이 불가능하다.

DSM 또한 $\nabla_{z_t} \log q(z_t)$가 C의 형태로 나타나기 때문에 뒤에 떨어 뭔지 모른다.

이런 점을 극복하기 위해 eq.5를 decompose.

Theorem 1.

주어진 두 분포 $q(z_0|x), p(z_0) \in \mathbb{R}^D$는 시간 t에서의 $q(z_t|x)$, $p(z_t)$의 marginal distribution

$\log q(z_t|x)$와 $\log p(z_t)$를 smooth condition을 가정하면.

$$CE(q(\mathbf{z}_0|\mathbf{x})||p(\mathbf{z}_0)) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[\frac{g(t)^2}{2}\mathbb{E}_{q(\mathbf{z}_t,\mathbf{z}_0|\mathbf{x})}\left[||\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t|\mathbf{z}_0) - \nabla_{\mathbf{z}_t}\log p(\mathbf{z}_t)||_2^2\right]\right] + \frac{D}{2}\log\left(2\pi e\sigma_0^2\right),$$

↳ $q(z_t, z_0|x) = q(z_t|z_0)q(z_0|x)$

Normal transition kernel $q(z_t|z_0) = N(z_t; \mu_t(z_0), \sigma_t^2 I)$

↳ obtain from $f(t), g(t)$

↳ 증명 : Appendix A.

eq.4와 다르게, $\nabla_{z_t} \log q(z_t)$의 의존 x

↳ DSM 사용가능 . $p(z_0)$ optim과 $q(z_0|x)$ encoding 에도 사용가능

↳ 복잡한 분포에서도 가능

Theorem 1에서 $\nabla_{z_t} \log p(z_t)$는 $p(z_0)$ 분포로 diffusion 해서 생기는 score.

- Mixing Normal and Neural Score Functions

VAE나, 최근 계층적 prior 모델들도, prior를 $N(z_0; 0, I)$로 가정한다.

single-D latent space를 가정하면, time t에서의 mixture는

$$p(z_t) \propto N(z_t; 0, I)^{1-\alpha} \underbrace{p'_\theta(z_t)^\alpha}$$

↳ trainable SGM prior, α: learnable scalar.

이점:

1) VAE pretrain 시에, $\alpha=0$으로 설정하면 됨

pretrain에서 prior를 $N(z_0; 0, I)$로 만든것처럼 SGM이 simple distribution을 학습하게 됨.

2) mixture의 score function은

$$\nabla_{z_t} \log p(z_t) = -(1-\alpha) z_t + \alpha \nabla_{z_t} \log p'_\theta(z_t)$$

↳ α가 작을 때 linear에 의존하기 때문에 reverse SDE가 빨리 풀림.

$z_t \sim p(z_t|z_0)$에서 $z_t = \mu_t(z_0) + \sigma_t \epsilon$ 일 때,

$\nabla_{z_t} \log q(z_t|z_0) = -\epsilon/\sigma_t$ ($\simeq$ DDPM) ⇒ ?

score function $\nabla_{z_t} \log p(z_t) := -G_\theta(z_t, t)/\sigma_t$

↳ $G_\theta(z_t, t) := \sigma_t(1-\alpha) \odot z_t + \alpha \odot \epsilon'_\theta(z_t, t)$

⇓ CE 단순화

$$\mathrm{CE}(q_\phi(\mathbf{z}_0|\mathbf{x})||p_\theta(\mathbf{z}_0)) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[\frac{w(t)}{2}\mathbb{E}_{q_\phi(\mathbf{z}_t,\mathbf{z}_0|\mathbf{x}),\epsilon}\left[||\epsilon-\epsilon_\theta(\mathbf{z}_t,t)||_2^2\right]\right] + \frac{D}{2}\log\left(2\pi e\sigma_0^2\right),$$

↳ $w(t) = g(t)/\sigma_t^2$

- Training with different weight Mechanism

w(t)의 weight mechanism은 3가지 정도 된다.

Table 1: Weighting mechanisms

Mechanism	Weights
Weighted	$w_{\text{ll}}(t) = g(t)^2/\sigma_t^2$
Unweighted	$w_{\text{un}}(t) = 1$
Reweighted	$w_{\text{re}}(t) = g(t)^2$

→ maximum likelihood
→ DDPM
→ reweight

encoder $q(z_0|x)$는 $p(z_0|x)$의 가까워지도록 학습된다.

보통 Unweighted의 비슷한 분포.

variance reduction이 더 안정된다, 등의 장점.

Training Objective

$$\min_{\phi,\psi} \mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[-\log p_\psi(\mathbf{x}|\mathbf{z}_0)\right] + \mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[\log q_\phi(\mathbf{z}_0|\mathbf{x})\right] + \mathbb{E}_{t,\boldsymbol{\epsilon},q(\mathbf{z}_t|\mathbf{z}_0),q_\phi(\mathbf{z}_0|\mathbf{x})}\left[\frac{w_{\text{ll}}(t)}{2}||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\mathbf{z}_t,t)||_2^2\right] \quad (8)$$

$$\min_{\theta} \mathbb{E}_{t,\boldsymbol{\epsilon},q(\mathbf{z}_t|\mathbf{z}_0),q_\phi(\mathbf{z}_0|\mathbf{x})}\left[\frac{w_{\text{ll/un/re}}(t)}{2}||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\mathbf{z}_t,t)||_2^2\right] \quad \text{with} \quad q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t;\boldsymbol{\mu}_t(\mathbf{z}_0),\sigma_t^2\mathbf{I}), \quad (9)$$

- Variance Reduction.

eq 8.9는 high variance를 가진다.

복잡 좋아기.

Variance preserving : $dz = -\frac{1}{2}\beta(t)z\,dt + \sqrt{\beta(t)}\,dw$, $\beta(t) = \beta_0 + (\beta_1-\beta_0)t$.

- Variance reduction for likelihood weighting

Appendix B에서, $q(z_0) = p(z_0) = N(z_0; 0, I)$ 일 때,

$$CE(q(z_0)\|p(z_0)) = \frac{D}{2}E_{t\sim U[0,1]}[d\log\sigma_t^2/dt] + \text{const.}$$

1) Geometric VPSDE.

uniform t에 대한 variance가 같아지면, $d\log\sigma_t^2/dt$가 $t\sim[0,1]$에 대해 uniform

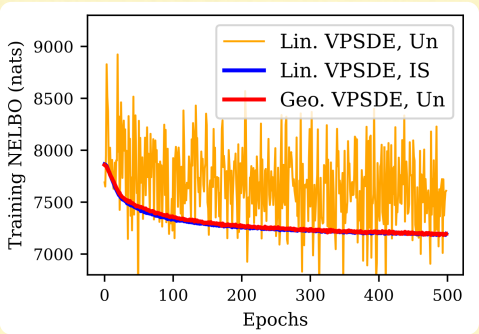
$\beta(t) = \log(\sigma_{max}^2/\sigma_{min}^2)\dfrac{\sigma_t^2}{(1-\sigma_t^2)}$, $\sigma_t^2 = \sigma_{min}^2(\sigma_{max}^2/\sigma_{min}^2)^t$ 일 때 만족.

VESPEC 에서처럼, t에 대한 reduction variance를 계산

2) Importance Sampling (IS)

$r(t) \propto d\log\sigma_t^2/dt$ 계산

$t = var^{-1}((\sigma_1^2)^p(\sigma_0^2)^{1-p})$



- Variance reduction for Unweight and reweight.

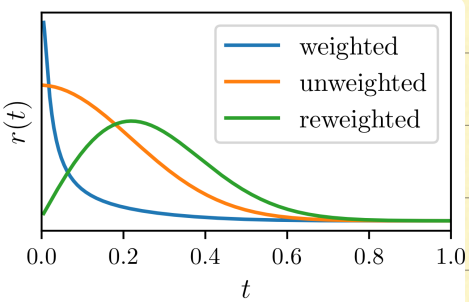


Figure 3: IS distributions

App B, G

* Related Work.

* Experiments.

NVAE + NCSN++

black box ODE solver for sampling

Table 2: Generative performance on CIFAR-10.

	Method	NLL↓	FID↓
Ours	LSGM (FID)	≤3.43	**2.10**
	LSGM (NLL)	≤**2.87**	6.89
	LSGM (balanced)	≤2.95	2.17
	VAE Backbone	2.96	43.18
VAEs	VDVAE [21]	**2.87**	-
	NVAE [20]	2.91	23.49
	VAEBM [76]	-	12.19
	NCP-VAE [56]	-	24.08
	BIVA [48]	3.08	-
	DC-VAE [77]	-	17.90
Score	NCSN [3]	-	25.32
	Rec. Likelihood [40]	3.18	9.36
	DSM-ALS [39]	3.65	-
	DDPM [1]	3.75	3.17
	Improved DDPM [26]	2.94	11.47
	SDE (DDPM++) [2]	2.99	2.92
	SDE (NCSN++) [2]	-	2.20
Flows	VFlow [19]	2.98	-
	ANF [18]	3.05	-
Aut. Reg.	DistAug aug [78]	2.53	42.90
	Sp. Transformers [79]	2.80	-
	δ-VAE [80]	2.83	-
	PixelSNAIL [81]	2.85	-
	PixelCNN++ [82]	2.92	-
GANs	AutoGAN [83]	-	12.42
	StyleGAN2-ADA [84]	-	2.92

Table 3: Generative results on CelebA-HQ-256.

	Method	NLL↓	FID↓
Ours	LSGM	≤**0.70**	**7.22**
	VAE Backbone	**0.70**	30.87
VAEs	NVAE [20]	**0.70**	29.76
	VAEBM [76]	-	20.38
	NCP-VAE [56]	-	24.79
	DC-VAE [77]	-	15.80
Score	SDE [2]	-	**7.23**
Flows	GLOW [85]	1.03	68.93
Aut. Reg.	SPN [86]	0.61	-
GANs	Adv. LAE [87]	-	19.21
	VQ-GAN [64]	-	10.70
	PGGAN [88]	-	8.03

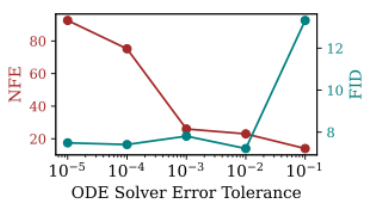


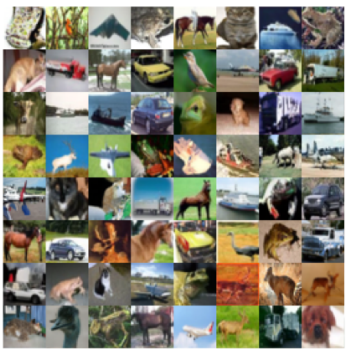
Figure 4: FID and number of function evaluations (NFEs) for different ODE solver error tolerances on CelebA-HQ-256. LSGM takes 4.15 sec. for sampling while the original SGM [2] takes 45 min. with PC and 3.9 min. with ODE-based sampling.

Table 4: Dyn. binarized OMNIGLOT results.

	Method	NELBO↓	NLL↓
Ours	LSGM	**87.79**	≤**87.79**
VAEs	NVAE [20]	93.92	90.75
	BIVA [48]	93.54	91.34
	DVAE++ [51]	-	92.38
	Ladder VAE [90]	-	102.11
Aut. Reg.	VLVAE [47]	-	89.83
	VampPrior [59]	-	89.76
	PixelVAE++ [91]	-	88.29

Table 5: Dynamically binarized MNIST results.

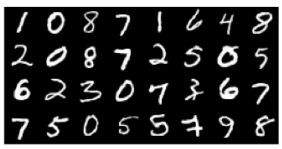
	Method	NELBO↓	NLL↓
Ours	LSGM	**78.47**	≤78.47
VAEs	NVAE [20]	79.56	**78.01**
	BIVA [48]	80.06	78.41
	IAF-VAE [24]	80.80	79.10
	DVAE++ [51]	-	78.49
Aut. Reg.	PixelVAE++ [91]	-	78.00
	VampPrior [59]	-	78.45
	MAE [92]	-	77.98



(a) CIFAR-10

(b) CelebA-HQ-256

(c) OMNIGLOT



(d) MNIST

Figure 5: Generated samples for different datasets. For binary datasets, we visualize the decoder mean. LSGM successfully generates sharp, high-quality, and diverse samples (additional samples in appendix).

Table 6: Ablations on SDEs, objectives, weighting mechanisms, and variance reduction. Details in App. G.

SGM-obj.-weighting		w_{ll}		w_{un}				w_{re}			
t-**sampling (SGM-obj.)**		$\mathcal{U}[0,1]$	$r_{\text{ll}}(t)$	$\mathcal{U}[0,1]$		$r_{\text{un}}(t)$		$\mathcal{U}[0,1]$		$r_{\text{re}}(t)$	
t-**sampling (q-obj.)**		*rew.*	*rew.*	*rew.*	$r_{\text{ll}}(t)$	*rew.*	$r_{\text{ll}}(t)$	*rew.*	$r_{\text{ll}}(t)$	*rew.*	$r_{\text{ll}}(t)$
Geom.-VPSDE	**FID**↓	10.18	*n/a*	*NaN*	*NaN*	*n/a*	*n/a*	22.21	*NaN*	7.29	7.18
	NELBO↓	2.96	*n/a*	*NaN*	*NaN*	*n/a*	*n/a*	3.04	*NaN*	2.99	2.99
VPSDE	**FID**↓	6.15	8.00	*NaN*	*NaN*	5.39	5.39	*NaN*	4.99	15.12	6.19
	NELBO↓	2.97	2.97	*NaN*	*NaN*	2.98	2.98	*NaN*	2.99	3.03	2.99

* Conclusion.

latent 에서 diffusion model

↳ more representation.

↳ discrete image modeling, smooth SGM.

Contribution.

1) 새로운 CE objective 제안

2) Mixing Prior 와 prior parameterization

3) variance reduction

SGM보다 빠른 sampling