Deep Generative Learning via Variational Gradient Flow

Yuan Gao [1] Yuling Jiao [2] Yang Wang [3] Yao Wang [4] Can Yang [3] Shunkang Zhang [3]

Abstract

We propose a framework to learn deep generative models via **V**ariational **Gr**adient Fl**ow** (VGrow) on probability spaces. The evolving distribution that asymptotically converges to the target distribution is governed by a vector field, which is the negative gradient of the first variation of the f-divergence between them. We prove that the evolving distribution coincides with the pushforward distribution through the infinitesimal time composition of residual maps that are perturbations of the identity map along the vector field. The vector field depends on the density ratio of the pushforward distribution and the target distribution, which can be consistently learned from a binary classification problem. Connections of our proposed VGrow method with other popular methods, such as VAE, GAN and flow-based methods, have been established in this framework, gaining new insights of deep generative learning. We also evaluated several commonly used divergences, including Kullback-Leibler, Jensen-Shannon, Jeffreys divergences as well as our newly discovered "logD" divergence which serves as the objective function of the logD-trick GAN. Experimental results on benchmark datasets demonstrate that VGrow can generate high-fidelity images in a stable and efficient manner, achieving competitive performance with state-of-the-art GANs.

1. Introduction

Learning the generative model, i.e., the underlying data generating distribution, based on large amounts of data is a fundamental task in machine learning and statistics (Salakhutdinov, 2015). Recent advances in deep generative models have provided novel techniques for unsupervised and semi-supervised learning, with broad applications varying from image synthesis (Reed et al., 2016), semantic image editing (Zhu et al., 2016), image-to-image translation (Zhu et al., 2017) to low-level image processing (Ledig et al., 2017). Implicit deep generative models are extremely powerful and flexible to approximate the target distribution by learning deep samplers (Mohamed & Lakshminarayanan, 2016) including *generative adversarial networks* (GAN) (Goodfellow et al., 2014) and likelihood-based models, such as *variational auto-encoders* (VAE) (Kingma & Welling, 2014) and *flow-based methods* (Dinh et al., 2015), as the representatives. The above-mentioned implicit deep generative models focus on learning a deterministic or stochastic nonlinear mapping that try to transform low-dimensional latent samples from a simple reference distribution to samples that closely match the target distribution.

GAN builds a minmax two player game between the generator and the discriminator. During training, the generator transforms samples from a simple reference distribution into samples that would hopefully deceive the discriminator, while the discriminator conducts a differential two-sample test to distinguish the generated samples from the observed samples. The objective of vanilla GAN amounts to the *Jensen-Shannon* (JS) divergence between the learned distribution and the target distribution. Vanilla GAN generates sharp image samples but suffers from the instability issue (Arjovsky et al., 2017). A myriad of extensions to vanilla GAN have been investigated, either theoretically or empirically, in order to achieve a stable training and high-quality sample generation. Existing work includes but is not limited to designing new learning procedures or network architectures (Denton et al., 2015; Radford et al., 2015; Zhang et al., 2017; Zhao et al., 2017; Arora et al., 2017; Tao et al., 2018; Brock et al., 2018), seeking alternative distribution discrepancy measures as loss criteria in the feature or data space (Li et al., 2015; Dziugaite et al., 2015; Li et al., 2017; Sutherland et al., 2017; Bińkowski et al., 2018; Arjovsky et al., 2017; Mao et al., 2017; Mroueh & Sercu, 2017), exploiting

[1] School of Mathematics and Statistics, Xi'an Jiaotong University, China [2] School of Statistics and Mathematics, Zhongnan University of Economics and Law, China and KLATASDS-MOE, School of Statistics, East China Normal University, China [3] Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong [4] School of Management, Xi'an Jiaotong University, China. Correspondence to: Yuling Jiao <yulingjiaomath@whu.edu.cn>, Can Yang <macyang@ust.hk>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

insightful regularization methods (Che et al., 2017; Gulrajani et al., 2017; Miyato et al., 2018; Zhang et al., 2018), and building hybrid models (Donahue et al., 2017; Tolstikhin et al., 2017; Dumoulin et al., 2017; Ulyanov et al., 2018; Huang et al., 2018).

VAE approximately minimizes the *Kullback-Leibler* (KL) divergence between the transformed distribution and the target distribution via optimizing a surrogate loss, i.e., the negative evidence lower bound defined as the reconstruction loss plus the regularization loss (Kingma & Welling, 2014). VAE enjoys optimization stability but was disputed for generating blurry image samples caused by the Gaussian decoder and the marginal log-likelihood based loss (Tolstikhin et al., 2017). Adversarial auto-encoders (Makhzani et al., 2016) use GAN to penalize the discrepancy between the aggregated posterior of latent codes and the simple prior distribution. Wasserstein auto-encoders (Tolstikhin et al., 2018) extend adversarial auto-encoders to general penalized optimal transport objectives (Bousquet et al., 2017) to alleviate the blurriness. Similar ideas are found in some works on disentangled representations of natural images (Higgins et al., 2017; Kumar et al., 2018).

Flow-based methods minimize exactly the negative log-likelihood, i.e., the KL divergence, where the model density is the pushforward density of a simple reference distribution through a sequence of learnable invertible transformations called normalizing flows (Rezende & Mohamed, 2015). The research on flow-based generative models mainly focuses on designing neural network architectures to trade off the representation power and the computation complexity of log-determinants (Dinh et al., 2015; 2017; Kingma et al., 2016; Papamakarios et al., 2017; Alessio et al., 2018; Kingma & Dhariwal, 2018).

In this paper, we propose a general framework to learn a deep generative model via combining the strengths of variational gradient flow (VGrow) on probability spaces, particle optimization and deep neural networks. Our method aims to find a deterministic transport map that transforms low-dimensional samples from a simple reference distribution, such as the standard normal distribution or the uniform distribution, into samples from the target distribution. The evolving distribution that asymptotically converges to the target distribution is governed by a vector field, which is the negative gradient of the first variation of the f-divergence between the evolving distribution and the target distribution. We prove that the evolving distribution coincides with the pushforward distribution through the infinitesimal time composition of residual maps that are perturbations of the identity map along the vector field. At the population level, the vector field only depends on the density ratio of the pushforward distribution and the target distribution, which can be consistently learned from a binary classification problem to distinguish the observed data sampling from the target distribution from the generated data sampling from pushforward distribution. Both the transform and the binary classifier are parameterized with deep neural networks and trained via stochastic gradient descent (SGD). Connections of our proposed VGrow method with other popular methods, such as VAE, GAN and flow-based methods, have been established in our framework, gaining new insights of deep generative learning. We also evaluated several commonly used divergences, including KL, JS, Jeffreys divergences as well as our newly discovered "logD" divergence serving as the objective function of the logD-trick GAN, which is of independent interest of its own. We test VGrow with the above-mentioned four divergences on four benchmark datasets including MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), CIFAR10 (Krizhevsky & Hinton, 2009) and CelebA (Liu et al., 2015)[1]. The VGrow learning procedure is very stable, as indicated from our established theory. The resulting deep sampler obtains realistic-looking images, achieving competitive performance with state-of-the-art GANs.

2. Background, Notation and Theory

Let $\{\mathbf{X}_i\}_{i=1}^N \subset \mathbb{R}^d$ be i.i.d. samples from an unknown target distribution ν. We assume that ν admits the density p with respective to the Lebesgue measure. (All distributions hold the same assumption hereinafter.) Our aim is to learn the distribution ν via constructing variational gradient flow on the space of Borel probability measures $\mathcal{P}(\mathbb{R}^d)$. To this end, the following background studied by Ambrosio et al. (2008) is needed.

Given $\mu \in \mathcal{P}(\mathbb{R}^d)$ with the density q, we use the f-divergence (Ali & Silvey, 1966) to measure the discrepancy between μ and ν which is defined as

$$\mathbb{D}_f(q\|p) = \int p(\mathbf{x}) f\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}, \tag{1}$$

where $f : \mathbb{R}^+ \to \mathbb{R}$ is a convex and continuous function satisfying $f(1) = 0$. We also require $f(\cdot)$ is twice-differentiable. Let $\mathcal{F}[q]$ denote the energy functional $\mathbb{D}_f(\cdot\|p) : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}^+ \cup \{0\}$ for simplicity. Obviously $\mathcal{F}[q] \geq 0$ and $\mathcal{F}[q] = 0$ iff $q(\mathbf{x}) = p(\mathbf{x}), \forall \mathbf{x} \in \text{supp}(p) \cup \text{supp}(q)$.

Lemma 2.1. Let $\frac{\delta \mathcal{F}}{\delta q}(q) : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ denote the first variation of $\mathcal{F}[\cdot]$ at q, then $\left(\frac{\delta \mathcal{F}}{\delta q}(q)\right)(\mathbf{x}) = f'(r(\mathbf{x}))$ where $r(\mathbf{x}) = \frac{q(\mathbf{x})}{p(\mathbf{x})}$.

We consider a curve $\mu_t : \mathbb{R}^+ \to \mathcal{P}(\mathbb{R}^d)$ and μ_t admits the

[1] The code of VGrow is available at https://github.com/xjtuygao/VGrow.

density q_t. Let $\mathbf{v}_t = -\nabla\left(\frac{\delta\mathcal{F}}{\delta q_t}(q_t)\right) : \mathbb{R}^+ \to (\mathbb{R}^d \to \mathbb{R}^d)$ be the velocity vector field with $r_t(\mathbf{x}) = \frac{q_t(\mathbf{x})}{p(\mathbf{x})}$.

Definition. We call μ_t is a variational gradient flow of the energy functional $\mathcal{F}[\cdot]$ governed by the velocity vector field $\mathbf{v}_t$ if the following Vlasov-Fokker-Planck equation holds

$$\frac{d}{dt}q_t = -\nabla\cdot(q_t\mathbf{v}_t) \text{ in } \mathbb{R}^+ \times \mathbb{R}^d. \tag{2}$$

As shown in Lemma 2.2, the energy functional $\mathcal{F}[\cdot]$ decreases along the curve μ_t. As a consequence, the limit of q_t is the target p as $t \to \infty$.

Lemma 2.2.

$$\frac{d}{dt}\mathcal{F}[q_t] = -\mathbb{E}_{\mathbf{X}\sim q_t}[\|\mathbf{v}_t(\mathbf{X})\|^2]$$

At any fixed time $t \in \mathbb{R}^+$, let $\mathbf{X}$ be a random variable with the density q_t. Let $\mathbf{h}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^d$ be an element of the Hilbert space $\mathcal{H}(q_t) = [L^2(q_t)]^d$ and $s \in \mathbb{R}^+$ be a small positive number. Define a residual map $\mathbb{T}_{s,\mathbf{h}} : \mathbb{R}^d \to \mathbb{R}^d$ as a small permutation of the identify map $\mathbf{id}$ along $\mathbf{h}$, i.e.,

$$\mathbb{T}_{s,\mathbf{h}} = \mathbf{id} + s\mathbf{h}.$$

Let $\mathbb{T}_{s,\mathbf{h}}^{-1}$ be the inverse of $\mathbb{T}_{s,\mathbf{h}}$, which is well-defined when s is small enough. By the change of variables formula, the density of the pushforward distribution of $\mathbb{T}_{s,\mathbf{h}}(\mathbf{X})$ is governed by $(\mathbb{T}_{s,\mathbf{h}\#}q_t)(\mathbf{x}) = q_t(\mathbb{T}_{s,\mathbf{h}}^{-1}(\mathbf{x}))|\det(\nabla_{\mathbf{x}}\mathbb{T}_{s,\mathbf{h}}^{-1}(\mathbf{x}))|$. We use $\mathcal{L}[\mathbf{h}] = \mathbb{D}_f(\mathbb{T}_{s,\mathbf{h}\#}q_t\|p)$ to denote the functional of $\mathbf{h}$ mapping from $\mathcal{H}(q_t)$ to $\mathbb{R}^+ \cup \{0\}$. It is natural to find $\mathbf{h}$ satisfying $\mathcal{L}[\mathbf{h}] < \mathcal{L}[\mathbf{0}]$, which indicates the pushforward distribution $\mathbb{T}_{s,\mathbf{h}\#}q_t$ is much closer to p than q_t. We find such $\mathbf{h}$ via calculating the first variation of the functional $\mathcal{L}[\mathbf{h}]$ at $\mathbf{0}$.

Theorem 2.1. For any $\mathbf{g} \in \mathcal{H}(q_t)$, if the vanishing condition $\lim\limits_{\|\mathbf{x}\|\to\infty} \|f'(r_t(\mathbf{x}))q_t(\mathbf{x})\mathbf{g}(\mathbf{x})\| = 0$ is satisfied, then

$$\left\langle \frac{\delta\mathcal{L}}{\delta\mathbf{h}}[\mathbf{0}], \mathbf{g} \right\rangle_{\mathcal{H}(q_t)} = \langle f''(r_t)\nabla r_t, \mathbf{g}\rangle_{\mathcal{H}(q_t)}.$$

The vanishing condition assumed in Theorem 2.1 holds when the densities have compact supports or light tails. Theorem 2.1 shows that the residual map defined as a small perturbation of the identity map along the velocity vector field $\mathbf{v}_t$ can push samples from q_t into samples more likely sampled from p.

Theorem 2.2. The evolving distribution q_t under the infinitesimal pushforward map $\mathbb{T}_{s,\mathbf{v}_t}$ satisfies the Vlasov-Fokker-Planck equation (2).

As consequences of Theorem 2.2, we know the pushforward distribution through the residual maps with infinitesimal time perturbations is the same as the variational gradient flow. This connection motivates us to approximately solve the Vlasov-Fokker-Planck equation (2) via finding a pushforward map defined as an aggregate composition of discrete time residual maps with a small step size as long as we obtain the vector field $\mathbf{v}_t$. By definition, the vector field $\mathbf{v}_t$ is an explicit function of density ratio r_t, which is well-studied, see for example, (Sugiyama et al., 2012).

Lemma 2.3. Let $(\mathbf{X}, Y)$ be a random variable pair admitting $p(\mathbf{x}, y)$ with the binary random variable $Y \sim p(y)$ taking the value in $\{-1, +1\}$. Denote $q(\mathbf{x}) = p(\mathbf{x}|Y = -1)$, $p(\mathbf{x}) = p(\mathbf{x}|Y = 1)$ and $r(\mathbf{x}) = \frac{q(\mathbf{x})}{p(\mathbf{x})}$. Let $d^*(\mathbf{x}) = \arg\min\limits_{d(\mathbf{x})} \mathbb{E}_{(\mathbf{X},Y)\sim p(\mathbf{x},y)} \log(1 + \exp(-d(\mathbf{X})Y))$. If $p(Y = 1) = p(Y = -1)$, then $r(\mathbf{x}) = \exp(-d^*(\mathbf{x}))$.

According to Lemma 2.3, we can estimate the density ratio $r_t(\mathbf{x}) = \frac{q_t(\mathbf{x})}{p(\mathbf{x})}$ with samples. Let $\{\mathbf{Z}_i\}_{i=1}^N, \{\mathbf{X}_i\}_{i=1}^N$ be samples from $q_t(\mathbf{x})$ and $p(\mathbf{x})$, respectively. We introduce a random variable Y, and assign a label $Y_i = -1$ for $\mathbf{Z}_i$ and $Y_i = 1$ for $\mathbf{X}_i$. Define

$$\begin{aligned}\hat{d}(\mathbf{x}) = \arg\min_{d(\mathbf{x})} \sum_{i=1}^{N} \big(&\log(1 + \exp(-d(\mathbf{X}_i)) \\ &+ \log(1 + \exp(d(\mathbf{Z}_i))\big),\end{aligned} \tag{3}$$

then $\hat{r}(\mathbf{x}) = \exp(-\hat{d}(\mathbf{x}))$ consistently estimates $r_t(\mathbf{x})$ as $N \to \infty$.

3. Variational gradient flow (VGrow) learning procedure

With data $\{\mathbf{X}_i\}_{i=1}^N \subset \mathbb{R}^d$ sampled from an unknown target distribution $p(\mathbf{x})$, our goal is to learn a deterministic transport map that transforms low dimensional samples from a simple reference distribution such as a Gaussian distribution or a uniform distribution into samples from the underlying target $p(\mathbf{x})$.

To this end, we parameterize the sought transform via a deep neural network $G_\theta : \mathbb{R}^\ell \to \mathbb{R}^d$ with $\ell \ll d$, where θ denotes its parameters. We sample particles $\{\mathbf{W}_i\}_{i=1}^N$ from simple reference distribution and transform them into $\{\mathbf{Z}_i\}_{i=1}^N$ with the initial G_θ. We do the following two steps iteratively. First, we learn a density ratio by solving the optimization problem (3) with real data $\{\mathbf{X}_i\}_{i=1}^N$ and generated data $\{\mathbf{Z}_i\}_{i=1}^N$, where we parameterize $d(\cdot)$ into a neural network $D_\phi(\cdot)$. Then, we define a residual map $\hat{\mathbb{T}}$ using the estimated vector field with a small step size s and update $\{\mathbf{Z}_i\}_{i=1}^N$ through $\widehat{\mathbb{T}}(\cdot)$. According to the theory we discussed in Section 2, the above iteratively two steps can get particles $\{\mathbf{Z}_i\}_{i=1}^N$ more likely sampled from $p(\mathbf{x})$. So we can update the generator G_θ via fitting the pairs $\{(\mathbf{W}_i, \mathbf{Z}_i)\}_{i=1}^N$ and repeat the above whole procedure as desired with warm start. We give a detailed description of the VGrow learning procedure as follows.

- **Outer loop**
  - Sample $\{\mathbf{W}_i\}_{i=1}^N \subset \mathbb{R}^\ell$ from the simple reference distribution and let particles $\mathbf{Z}_i = G_\theta(\mathbf{W}_i), i = 1, 2, ..., N$.
  - **Inner loop**
    - ∗ Restrict $d(\cdot)$ in (3) be a neural network $D_\phi(\cdot)$ with parameter ϕ and solve (3) with SGD to get $\hat{r}(\mathbf{x}) = \exp(-D_\phi(\mathbf{x}))$.
    - ∗ Define the residual map $\widehat{\mathbb{T}} = \mathbf{id} + s\widehat{\mathbf{h}}$ with a small step size s, where $\widehat{\mathbf{h}}(\mathbf{x}) = -f''(\hat{r}(\mathbf{x}))\nabla\hat{r}(\mathbf{x})$.
    - ∗ Update the particles $\mathbf{Z}_i = \widehat{\mathbb{T}}(\mathbf{Z}_i), i = 1, 2, ..., N$.
  - **End inner loop**
  - Update the parameter θ of $G_\theta(\cdot)$ via solving $\min_\theta \sum_{i=1}^N \|G_\theta(\mathbf{W}_i) - \mathbf{Z}_i\|^2$ with SGD.
- **End outer loop**

We consider four divergences in this paper. The form of the four divergences and their second order derivatives are shown in Table 1. They are three commonly used divergences, including KL, JS and Jeffreys divergences, as well as our newly discovered "logD" divergence serving as the objective function of the logD-trick GAN, which to the best of our knowledge is a new result.

Theorem 3.1. At the population level, the logD-trick GAN (Goodfellow et al., 2014) minimizes the "logD" divergence $\mathbb{D}_f(q(\mathbf{x})\|p(\mathbf{x}))$, with $f(u) = (u+1)\log(u+1) - 2\log 2$, where $q(\mathbf{x})$ is the distribution of generated data.

Table 1. Four representative f-divergences

f-Div	$f(u)$	$f''(u)$
KL	$u \log u$	$\frac{1}{u}$
JS	$-(u+1)\log\frac{u+1}{2} + u\log u$	$\frac{1}{u(u+1)}$
logD	$(u+1)\log(u+1) - 2\log 2$	$\frac{1}{u+1}$
Jeffreys	$(u-1)\log u$	$\frac{u+1}{u^2}$

4. Related Work

We discuss connections between our proposed VGrow learning procedure and related work, such as VAE, GAN and flow-based methods.

VAE (Kingma & Welling, 2014) is formulated as maximizing a lower bound based on the KL divergence. Flow-based methods (Dinh et al., 2015; 2017) minimize the KL divergence between the target distribution and a model distribution, which is the pushforward distribution of a simple reference distribution through a sequence of learnable invertible transforms. These transforms are parameterized as specifically designed neural networks to facilitate computations of log-determinants (Dinh et al., 2015; 2017; Kingma et al., 2016; Papamakarios et al., 2017; Kingma & Dhariwal, 2018) and the training process leads to a maximum likelihood estimation. Our VGrow also learns a sequence of simple residual maps governed by the variational gradient flow in probability spaces, which is quite different from the flow-based methods in principle.

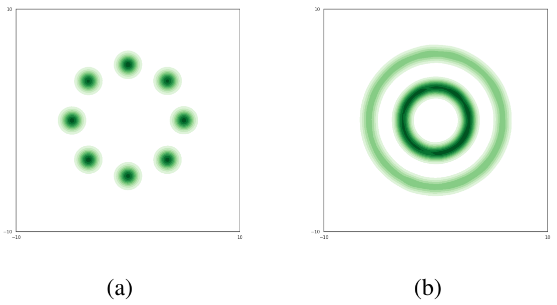
The vanilla GAN and the logD-trick GAN (Goodfellow et al., 2014) minimize the JS divergence and the "logD" divergence, respectively, as shown in Theorem 3.1. This idea can be extended to a general f-GAN (Nowozin et al., 2016), where f-divergences are utilized. Furthermore, based on f-divergences, GANs are formulated to solve the dual problem. In contrast, our VGrow directly minimizes the f-divergence from the primal form. The most related work of GANs to VGrow is (Johnson & Zhang, 2018; Nitanda & Suzuki, 2018; Wang & Liu, 2017), where functional gradient (first variation of functional) is adopted to favor the GAN training. Nitanda & Suzuki (2018) introduced a gradient layer based on first variation of generator loss in WGAN (Arjovsky et al., 2017) to accelerate convergence of training. In Wang & Liu (2017), a deep energy model was trained along Stein variational gradient (Liu & Wang, 2016), which was the projection of the first variation of KL divergence in Theorem 2.1 onto a reproducing kernel Hilbert space, please see the supplementary material for a proof. Johnson & Zhang (2018) proposed CFG-GAN that directly minimizes the KL divergence via functional gradient descent. In their work, the update direction is the gradient of log density ratio multiplied by a positive scaling function. They empirically set this scaling function to be 1 in their numerical study. Our VGrow is based on the general f-divergence, and Theorem 2.1 implies that the update direction in KL divergence case is indeed the gradient of log density ratio, and thus the scaling function of CFG-GAN should be exactly 1.

5. Experiments

First, two toy examples of fitting two-dimensional mixture distributions were conducted to illustrate the ability of VGrow to learn multimodal distributions without mode collapse. Next, we evaluated our model on four benchmark image datasets including MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), CIFAR10 (Krizhevsky & Hinton, 2009) and CelebA (Liu et al., 2015). We claim that all f-divergences with a twice-differentiable f are compatible with the general variational gradient flow (VGrow) framework, and four representatives in Table 1 were tested to demonstrate the effectiveness of VGrow for generative learning.

5.1. 2D toy examples

As shown in Figure 1, the first simulation data were generated from a mixture of eight two-dimensional Gaussians and the second were two concentric circles with Gaussian noise. Single hidden layer neural networks with ReLU activation functions were employed to parameterize the deep sampler and the deep classifier. In order to enhance the representational capacity of networks, the number of hidden neurons were set to 512, i.e., the dimension of layers was 2-512-2 for both the sampler and the classifier. We visualized the evolving particles by kernel density estimation (KDE) in Figure 2. VGrow transformed the standard normal distribution to capture all the modes of Gaussian mixture with only one hidden layer. Furthermore, VGrow provided good approximations to the modes and support of the target distribution with a few outer loops. With more steps taken, the distributions were fitted much better.



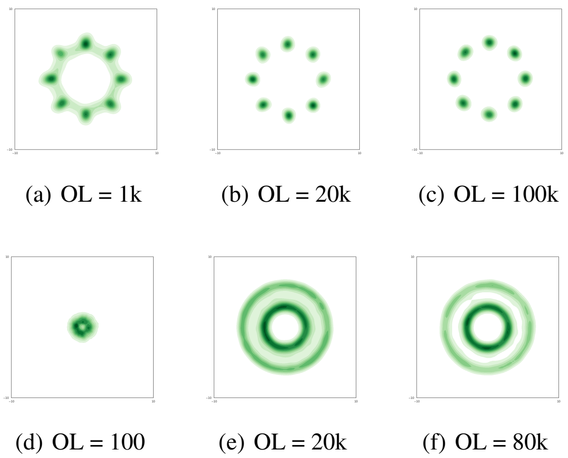
(a) (b)

Figure 1. KDE plots for (a) Mixture of Gaussians (MoG); (b) Concentric circles with Gaussian noise.

5.2. Experimental setup

f-divergences. Theoretically, our model works for the whole f-divergence family by simply plugging the twice-differentiable function f in. Special cases are obtained when specific f-divergences are considered. At the population level, when the KL divergence is adopted, our VGrow naturally gives birth to CFG-GAN while the adoption of JS divergence leads us to the vanilla GAN. As we proved above, GAN with the logD trick corresponds to our newly discovered ”logD” divergence which belongs to the f-divergence family. Moreover, we consider the Jeffreys divergence to show that our model is applicable to other f-divergences. We name these four cases VGrow-KL, VGrow-JS, VGrow-logD and VGrow-JF.

Datasets. We chose four benchmark datasets which included three small datasets (MNIST, FashionMNIST, CIFAR10) and one large dataset (CelebA) from GAN literature. Both MNIST and FashionMNIST have a training set of 60k examples and a test set of 10k examples as 28 $\times$ 28 bilevel images. CIFAR10 has a training set of 50k examples and a test set of 10k examples as 32 $\times$ 32 color images. There are naturally 10 classes on these three datasets. CelebA consists of more than 200k celebrity images which were randomly divided into a training set and a test set, and the division ratio is approximately $9:1$. For MNIST and FashionMNIST, the input images were resized to 32×32 resolution. We also pre-processed CelebA images by first taking a 160 $\times$ 160 central crop and then resizing to the 64 $\times$ 64 resolution. Only the training sets are used to train our models.



(a) OL = 1k (b) OL = 20k (c) OL = 100k

(d) OL = 100 (e) OL = 20k (f) OL = 80k

Figure 2. KDE plots of the evolving particles. The first and second rows show pictures concerning MoG and circles, respectively. OL denotes the number of outer loops hereinafter.

Evaluation metrics. *Inception Score* (IS) (Salimans et al., 2016), calculates the exponential mutual information $\exp(\mathbb{E}_{\mathbf{G}}\mathrm{KL}[p(\mathbf{C}|\mathbf{G})\|p(\mathbf{C})])$ where $p(\mathbf{c}|\mathbf{g})$ is the underlying distribution of the class $\mathbf{C}$ that the generated image $\mathbf{g}$ belongs to and $p(\mathbf{c})$ is the marginal class distribution across generated images (Barratt & Sharma, 2018). To estimate $p(\mathbf{c}|\mathbf{g})$ and $p(\mathbf{c})$, we trained specific classifiers on MNIST, FashionMNIST, CIFAR10 following Johnson & Zhang (2018) using pre-activation ResNet-18 (He et al., 2016). We evaluated IS over 50k generated images. *Fréchet Inception Distance* (FID) (Heusel et al., 2017) computes the Wasserstein-2 distance with summary statistics (mean μ and variance Σ) of real images $\mathbf{x}$s and generated images $\mathbf{g}$s in the feature space of the Inception-v3 model (Szegedy et al., 2016), i.e., FID $=\|\mu_{\mathbf{x}}-\mu_{\mathbf{g}}\|_2^2+\mathrm{Tr}(\Sigma_{\mathbf{x}}+\Sigma_{\mathbf{g}}-2(\Sigma_{\mathbf{x}}\Sigma_{\mathbf{g}})^{\frac{1}{2}})$. In our work, FID is reported with respect to the 10k test examples on MNIST, FashionMNIST and CIFAR10 with the tensorflow implementation. In a nutshell, higher IS and lower FID are better.

Network architectures and hyperparameter settings. We adopted a new architecture modified from the residual networks used in Miyato et al. (2018). The modifications were comprised of reducing the number of batch normalization layers and introducing spectral normalization in the deep sampler / generator. The architecture was shared across the three small datasets and most hyperparameters were shared across different divergences. More residual blocks, upsampling and downsampling are employed on CelebA. Implementation details can be found in the second and third section of the supplementary material.

5.3. Results

Through our experiment, We demonstrate empirically that (1) VGrow is very stable in the training phase, and that (2) VGrow can generate high-fidelity samples that are comparable to real samples both visually and quantitatively. Comparisons with the state-of-the-art GANs suggest the effectiveness of VGrow.

Stability. It has been shown that the binary classification loss poorly correlates with the generating performance for JS divergence based GAN models (Arjovsky et al., 2017). We observed similar phenomena with our f-divergence based VGrow model, i.e., the classification loss changed a little at the beginning of training and then fluctuated around a constant value. Since the classfication loss was not meaningful enough to measure the generating performance, we turned to utilize the aforementioned inception score to draw IS-OL learning curves on MNIST, FashionMNIST and CIFAR10. The results are presented in Figure 3. As indicated in all three subfigures, the IS-OL learning curves are very smooth and the inception scores nearly monotonically increase until 3500 outer loops (almost 75 epochs) on MNIST and FashionMNIST as well as 4500 outer loops (almost 100 epochs) on CIFAR10.

Effectiveness. First, we list the real images and generated examples of our VGrow-KL model on the four benchmark datasets in Figure 4. We claim that the realistic-looking generated images are visually comparable to real images sampled from the training set. It is easy to distinguish which class the generated example belongs to even on CIFAR10. Second, Table 2 presents the FID scores for the considered four models, and the FID values on 10k training data of MNIST and FashionMNIST. Scores of generated samples are very close to scores on real data. Especially, VGrow-JS obtains average scores of 3.32 and 8.75 while the scores on training data are 2.12 and 4.16 on MNIST and FashionMNIST, respectively. Third, Table 3 shows FID evaluations of our four models, and the referred evaluations of state-of-the-art WGANs and MMDGANs from Arbel et al. (2018) based on 50k samples. Our VGrow-logD attain a score of 28.8 with less variance that is competitive with the best (28.5) of referred baseline evalution. VGrow-JS and VGrow-KL achieve better performance than the remaining referred baselines. In a word, the quantitative results in Table 2 and Table 3 illustrate the effectiveness of our VGrow model.

Table 2. Mean (standard deviation) of FID evaluations over 10k generated MNIST / FashionMNIST images with five-time bootstrap sampling. The last row states statistics of the FID scores between 10k training examples and 10k test examples.

Models	MNIST(10k)	FashionMNIST (10k)
VGrow-KL	3.66 (0.09)	9.30 (0.09)
VGrow-JS	**3.32 (0.05)**	**8.75 (0.06)**
VGrow-logD	3.64 (0.05)	9.51 (0.09)
VGrow-JF	3.40 (0.07)	9.72 (0.06)
Training set	2.12 (0.02)	4.16 (0.03)

Table 3. Mean (standard deviation) of FID evaluations over 50k generated CIFAR10 images with five-time bootstrap sampling. The last four rows are baseline results adapted from Arbel et al. (2018).

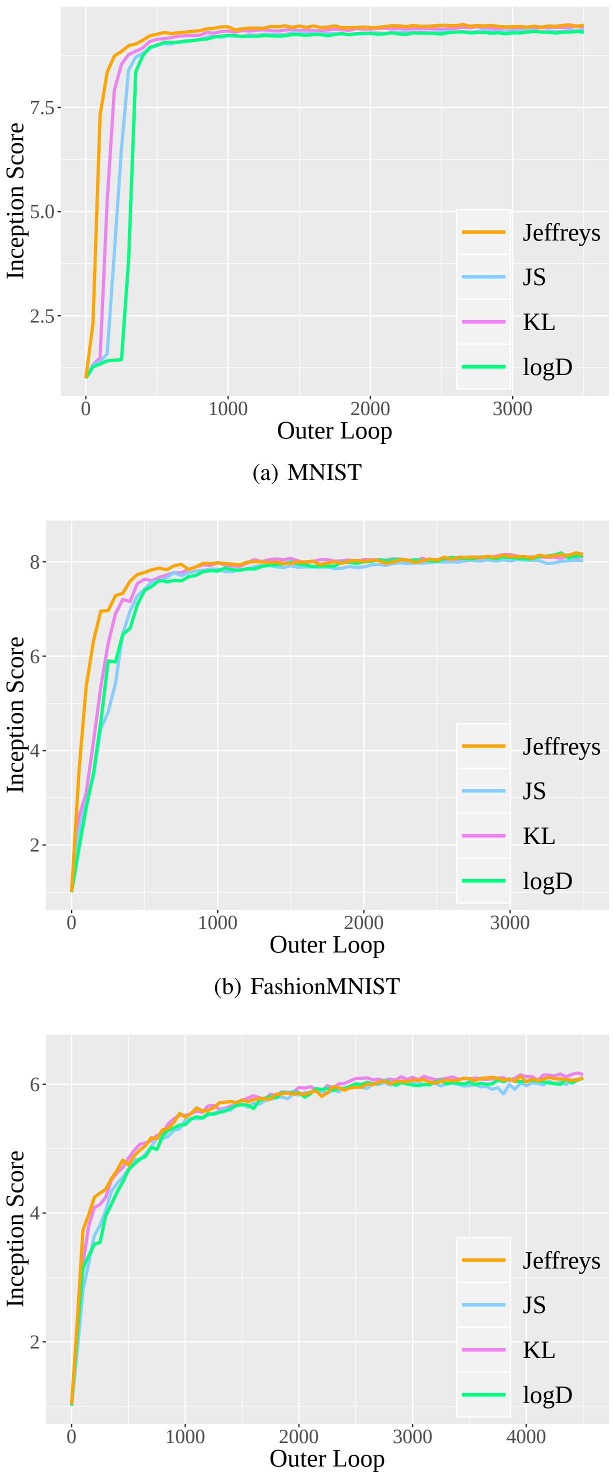
Models	CIFAR10 (50k)
VGrow-KL	29.7 (0.1)
VGrow-JS	29.1 (0.1)
VGrow-logD	**28.8 (0.1)**
VGrow-JF	32.3 (0.1)
WGAN-GP	31.1 (0.2)
MMDGAN-GP-L2	31.4 (0.3)
SMMDGAN	31.5 (0.4)
SN-SWGAN	**28.5 (0.2)**

6. Conclusion

We propose the VGrow framework to learn deep generative models. We discuss connections of our proposed VGrow method with VAE, GAN and flow-based methods. We evaluated VGrow on several divergences, including a newly discovered "logD" divergence which serves as the objective function of the logD-trick GAN. Experimental results on benchmark datasets demonstrate that VGrow can generate high-fidelity images in a stable and efficient manner, achieving competitive performance with state-of-the-art GANs.

Acknowledgement

We thank the reviewers for their helpful comments. This work was supported in part by the National Science Funding of China [11871474, 61701547, 11501440, 61501389]; the Hong Kong Research Grant Council [22302815, 12316116, 12301417 and 16307818]; The Hong Kong University of Science and Technology [R9405 and IGN17SC02]; and the research fund of KLATASDS-MOE.

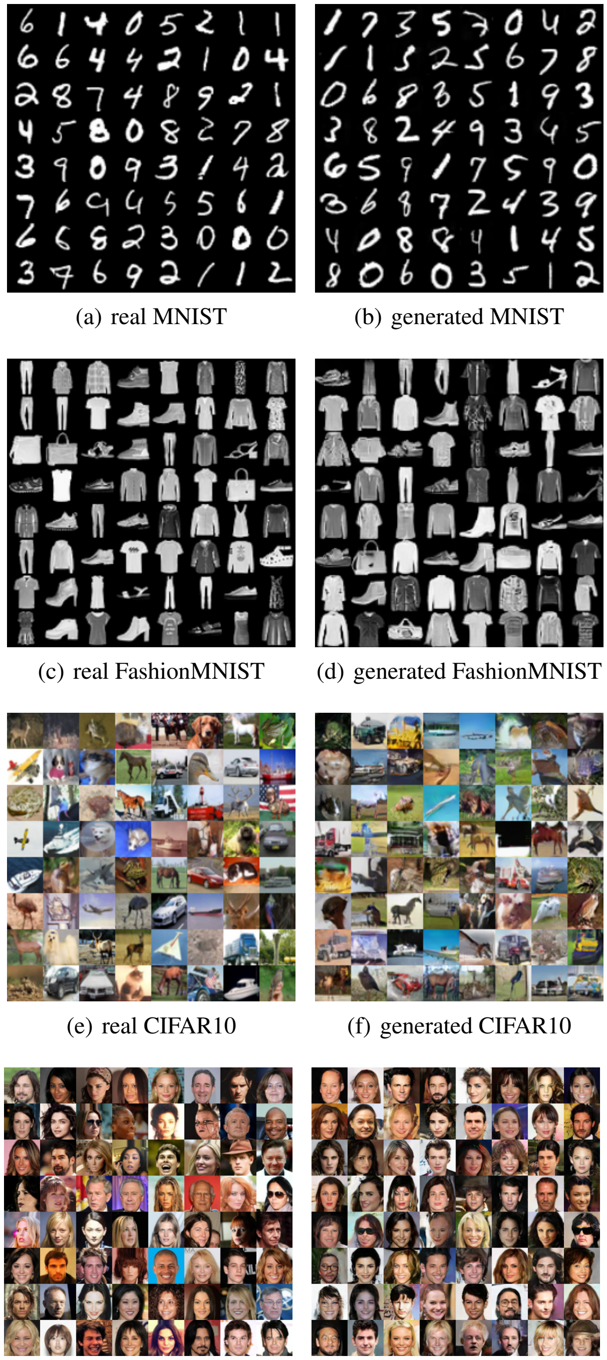


(a) MNIST

(b) FashionMNIST

(c) CIFAR10

Figure 3. IS-OL learning curves on MNIST, FashionMNIST and CIFAR10. The training of VGrow is very stable until 3500 outer loops on MNIST and FashionMNIST (4500 outer loops on CIFAR10).



(a) real MNIST

(b) generated MNIST

(c) real FashionMNIST

(d) generated FashionMNIST

(e) real CIFAR10

(f) generated CIFAR10

(g) real CelebA

(h) generated CelebA

Figure 4. Real samples and generated samples obtained by VGrow-KL on MNIST, FashionMNIST, CIFAR10 and CelebA.

References

Alessio, S., Bigoni, D., and Marzouk, Y. Inference via low-dimensional couplings. *Journal of Machine Learning Research*, 19(1):2639–2709, 2018.

Ali, S. M. and Silvey, S. D. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

Ambrosio, L., Gigli, N., and Savaré, G. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

Arbel, M., Sutherland, D., Bińkowski, M., and Gretton, A. On gradient regularizers for MMD GANs. In *NeurIPS*, 2018.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *ICML*, 2017.

Arora, S., Ge, R., Liang, Y., Ma, T., and Zhang, Y. Generalization and equilibrium in generative adversarial nets (GANs). In *ICML*, 2017.

Barratt, S. and Sharma, R. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying MMD GANs. In *ICLR*, 2018.

Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C.-J., and Schoelkopf, B. From optimal transport to generative modeling: the VEGAN cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. Mode regularized generative adversarial networks. In *ICLR*, 2017.

Denton, E. L., Chintala, S., szlam, a., and Fergus, R. Deep generative image models using a Laplacian pyramid of adversarial networks. In *NIPS*, 2015.

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear independent components estimation. In *ICLR*, 2015.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. In *ICLR*, 2017.

Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial feature learning. In *ICLR*, 2017.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially learned inference. In *ICLR*, 2017.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NIPS*, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of Wasserstein gans. In *NIPS*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *ECCV*, 2016.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. β-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Huang, H., Li, Z., He, R., Sun, Z., and Tan, T. IntroVAE: Introspective variational autoencoders for photographic image synthesis. In *NeurIPS*. 2018.

Johnson, R. and Zhang, T. Composite functional gradient learning of generative adversarial models. In *ICML*, 2018.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *NIPS*, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. In *ICLR*, 2018.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. MMD GAN: Towards deeper understanding of moment matching network. In *NIPS*, 2017.

Li, Y., Swersky, K., and Zemel, R. Generative moment matching networks. In *ICML*, 2015.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NIPS*, 2016.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *ICCV*, 2015.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. In *ICLR*, 2016.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. Least squares generative adversarial networks. In *ICCV*, 2017.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *ICML*, 2018.

Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Mroueh, Y. and Sercu, T. Fisher GAN. In *NIPS*, 2017.

Nitanda, A. and Suzuki, T. Gradient layer: Enhancing the convergence of adversarial training for generative models. In *AISTATS*, 2018.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *NIPS*, 2017.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *ICML*, 2015.

Salakhutdinov, R. Learning deep generative models. *Annual Review of Statistics and Its Application*, 2:361–385, 2015.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs. In *NIPS*, 2016.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Sutherland, D. J., Tung, H.-Y., Strathmann, H., De, S., Ramdas, A., Smola, A., and Gretton, A. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR*, 2017.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Tao, C., Chen, L., Henao, R., Feng, J., and Duke, L. C. Chi-square generative adversarial network. In *ICML*, 2018.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. In *ICML*, 2018.

Tolstikhin, I. O., Gelly, S., Bousquet, O., Simon-Gabriel, C.-J., and Schölkopf, B. AdaGAN: Boosting generative models. In *NIPS*, 2017.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. It takes (only) two: Adversarial generator-encoder networks. In *AAAI*, 2018.

Wang, D. and Liu, Q. Learning to draw samples: With application to amortized MLE for generative adversarial learning. In *ICLR workshop*, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

Zhao, J., Mathieu, M., and LeCun, Y. Energy-based generative adversarial network. In *ICLR*, 2017.

Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A. A. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.