Variational Bayesian GAN

Jen-Tzung Chien
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

Chun-Lin Kuo
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

***Abstract*—Generative adversarial network (GAN) has been successfully developing as a generative model where the artificial data drawn from the generator are misrecognized as real samples by a discriminator. Although GAN achieves the desirable performance, the challenge is that the mode collapse easily happens in the joint optimization of generator and discriminator. This study copes with this challenge by improving the model regularization by means of representing the weight uncertainty in GAN. A new Bayesian GAN is formulated and implemented to learn a regularized model from diverse data where the strong modes are flattened via the marginalization and the issues of model collapse and gradient vanishing are alleviated. In particular, we present a variational GAN (VGAN) where the encoder, generator and discriminator are jointly estimated according to the variational Bayesian inference. The experiments on image generation over two tasks (MNIST and CeleA) demonstrate the superiority of the proposed VGAN to the variational autoencoder, the standard GAN and the Bayesian GAN based on the sampling method. The learning efficiency and generation performance are evaluated.**

***Index Terms*—generative adversarial networks, Bayesian learning, variational autoencoder, computer vision**

I. Introduction

Generative adversarial network (GAN) [1]–[10] is known a generative model which allows approximate estimation of new data. GAN carries out the data generation procedure based on a two-player game between two neural networks. A generator model that synthesizes a data sample which can fool a discriminator model. Discriminator is trained to distinguish the synthesized sample from the true sample via an adversarial process. This approach avoids assuming the explicit data distribution and simply adopts the stochastic gradient descent training. The generative model through a learned GAN can directly serve as a density model of the training data. Sampling is simple for an efficient implementation. The network accepts the random noise as input and produces new samples in line with the observed training data. In the literature, the variational autoencoder (VAE) [11] was a well-known generative model which optimized the variational likelihood of training data and led to the meaningful reconstruction based on a lower bound of log likelihood for fitting model to data. However, new images synthesized by VAE were blurry. The regularization issue in VAE was not well treated with the fixed model parameters. In [12], the Bayesian neural network was proposed by incorporating the probabilistic weights in training of neural network parameters. Such an approach can produce the probabilistic guarantees on prediction performance through the learned parameters of weights from training samples. In general, the neural network is powerful to approximate a universal continuous function while the probabilistic model allows the uncertainty modeling of parameters for data generation.

This paper presents a new Bayesian framework for GAN which is motivated by the integrated idea from GAN, VAE and Bayesian neural network. By maximizing the variational lower bound of log likelihood under the setting of GAN, we develop a variational GAN which is capable of exploring the posterior over parameters as well as generating the realistic synthesized samples. The uncertainty of parameters is considered to capture different modes in data manifold. Such a variational GAN provides a more compact and efficient realization than the Bayesian GAN [13], which was implemented by sampling method from multiple parameter sets. Experiments on image generation are conducted to evaluate the performance of the varational Bayesian GAN with respect to other GANs.

II. Adversarial and Bayesian Learning

A. Generative adversarial network

Generative adversarial network (GAN) conducts an adversarial learning for a generator and a discriminator. The objective of generator is to synthesize the samples resembling real samples while the objective of discriminator is to distinguish real samples from synthesized ones. Let $\mathbf{x}$ be a real sample drawn from data distribution $p(\mathbf{x})$ (or $p_{\text{data}}(\mathbf{x})$) and $\mathbf{z}$ be a random noise sample from an arbitrary distribution $p(\mathbf{z})$. In a vanilla GAN, $p(\mathbf{z})$ is from a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Let G and D be the generator and discriminator, respectively. The generator takes $\mathbf{z}$ as an input and would like to produce an output sample $\widehat{\mathbf{x}} = G(\mathbf{z})$ which has the same distribution as $\mathbf{x}$. Denote the distribution of generator $G(\mathbf{z})$ as $p_{\text{gen}}(\mathbf{x})$. The discriminator D estimates the probability that an input sample is drawn from $p(\mathbf{x})$ rather than $p_{\text{gen}}(\mathbf{x})$. Ideally, $D(\mathbf{x}) = 1$ if $\mathbf{x} \sim p(\mathbf{x})$ and $D(\mathbf{x}) = 0$ if $\mathbf{x} \sim p_{\text{gen}}(\mathbf{x})$. Construction of GAN corresponds to run a minimax optimization to fulfill a two-player game for a joint training in accordance with $V(G, D)$

$$\min_G \max_D \; \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_{\text{gen}}(\mathbf{x})}[\log(1 - D(\mathbf{x}))] \tag{1}$$

which is seen as the negative cross entropy error function for a binary classification problem. $D(\mathbf{x})$ reflects a posterior probability of classifying a sample $\mathbf{x}$. Basically, an optimal discriminator is estimated to achieve the worst performance when classifying the samples from $p_{\text{data}}(\mathbf{x})$ and $p_{\text{gen}}(\mathbf{x})$.

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

Therefore, the maximization over $V(G,D)$ is run to train the discriminator while the minimization is performed to estimate the generator. Such a minimax optimization encourages the generator G to produce the sample $\mathbf{x}$ to fit $p(\mathbf{x})$ so as to fool discriminator D with the generated samples. Generator G and discriminator D are constructed by using adversarial learning based on the fine-tuning from the binary classification of the samples either from true distribution $p(\mathbf{x})$ or from prior distribution $p(\mathbf{z})$. Both G and D are trained via the backpropagation algorithm. Basically, GAN is seen as a deterministic machine where the uncertainties in G and D are disregarded.

In general, GAN performs well in characterizing the local structure in generated samples but may not catch precisely the global structure or the data distribution. In [14], GAN was combined with the variational autoencoder (VAE) [11] to compensate this weakness and balance the characterization between local and global features. Such a hybrid VAE and GAN was constructed by an encoder, a generator and a discriminator which were jointly trained according to the adversarial loss as well as the reconstruction loss. However, the pixel-based metrics might lead to substantial loss when little perturbation happened. The hybrid VAE and GAN was then improved by replacing the element-wise reconstruction error using the similarity metric learned from the discriminator. Basically, the variational inference in VAE or hyrbid VAE and GAN was performed by exploring the stochastic latent feature $\mathbf{z}$. The uncertainties of weight parameters in encoder, decoder and discriminator were *not* considered. Regularization issue was not sufficiently tackled. The latent code $\mathbf{z}$ is inferred by a trained encoder using the variational distribution $\mathbf{z} \sim q_\eta(\mathbf{z}|\mathbf{x})$ rather than simply sampled from the standard Gaussian $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ in GAN. Here, η denotes the parameter of encoder. Given the inferred samples $\mathbf{z}$, the decoder is seen as a generator which generates fake samples where the discriminator could not tell the difference from true samples.

B. Bayesian GAN

Although the synthesized samples using GAN are convincing, it is still difficult to generate the diverse examples in which the generator simply memorizes a few training examples to fool the discriminator. As a result, some modes are missing in generation process. To alleviate such a mode collapse problem [15], Bayesian GAN [13] was proposed to fulfill a fully probabilistic inference procedure where the *weight uncertainty* was taken into account in construction of GAN for data generation. The stochastic gradient Hamiltonian Monte Carlo (HMC) algorithm was implemented as a sampling approach to marginalize the posterior distributions over the weight parameters of generator θ_g and discriminator θ_d. Each mode in the posterior over generator weights basically represent the weight uncertainty from different generators. To infer the posterior over θ_g and θ_d from training samples $\mathbf{z} = \{\mathbf{z}_n\}_{n=1}^{N_g}$ and $\mathbf{x} = \{\mathbf{x}_n\}_{n=1}^{N_d}$, Bayesian GAN iteratively draws the samples from the conditional posteriors $p(\theta_g|\mathbf{z},\theta_d)$ and $p(\theta_d|\mathbf{z},\mathbf{x},\theta_g)$ by combining the likelihood function in Eq. (1) with the priors over parameters of generator and discriminator $p(\theta_g|\alpha_g)$ and $p(\theta_d|\alpha_d)$ with hyperparameters α_g and α_d with to yield

$$\left\{\prod_{n=1}^{N_g} D\big(G(\mathbf{z}_n,\theta_g),\theta_d\big)\right\}p(\theta_g|\alpha_g) \tag{2}$$

$$\left\{\prod_{n=1}^{N_d} D\big(\mathbf{x}_n,\theta_d\big)\prod_{n=1}^{N_g}\big(1-D\big(G(\mathbf{z}_n,\theta_g),\theta_d\big)\big)\right\}p(\theta_d|\alpha_d) \tag{3}$$

respectively. Practically, these posteriors are marginalized over noise $\mathbf{z}$ using the Monte Carlo method in a form of

$$p(\theta_g|\theta_d) = \int p(\theta_g|\mathbf{z},\theta_d)p(\mathbf{z}|\theta_d)d\mathbf{z} \approx \frac{1}{L_z}\sum_{l=1}^{L_z} p(\theta_g|\mathbf{z}^{(l)},\theta_d) \tag{4}$$

and $p(\theta_d|\mathbf{x},\theta_g) \approx \frac{1}{L_z}\sum_{l=1}^{L_z} p(\theta_d|\mathbf{z}^{(l)},\mathbf{x},\theta_g)$ for sampling procedure of θ_g and θ_d, respectively, where $\mathbf{z}^{(l)} \sim p(\mathbf{z})$ is the white noise sample with totally L_z samples. Finally, there are L_θ sets of parameters $\{\theta_g^{(l)},\theta_d^{(l)}\}_{l=1}^{L_\theta}$ drawn for prediction or generation of new data. The Bayesian GAN (denoted by BGAN) is therefore constructed.

III. Variational Bayesian GAN

BGAN basically compensates the weight uncertainty and partially mitigates the dilemma of model collapse. But, the sampling procedure is time-consuming and hard to converge. Instead of using the stochastic gradient HMC algorithm for posterior sampling, this paper presents the variational Bayesian inference for the posterior over parameter weights in a new generative model (called the variational GAN, VGAN).

A. Variational Bayesian inference

Similar to BGAN, there are two latent variables $\mathbf{z}$ and θ in VGAN, Following the variational inference, the marginalization of likelihood function over $\{\mathbf{z},\theta\}$ is first calculated and then used to find the variational lower bound $\mathcal{L}(\eta,\alpha)$ by

$$\begin{aligned}\log p(\mathbf{x}) &= \log\int\int p(\mathbf{x},\mathbf{z},\theta)\,d\mathbf{z}d\theta \\ &\geq \mathbb{E}_{q(\theta|\alpha)}\Bigg[\sum_{n=1}^{N}\Big\{\mathbb{E}_{q_\eta(\mathbf{z}_n|\mathbf{x}_n)}[\log p_\theta(\mathbf{x}_n|\mathbf{z}_n)] \\ &\qquad - \mathrm{KL}\big(q_\eta(\mathbf{z}_n|\mathbf{x}_n)\|p(\mathbf{z}_n)\big)\Big\}\Bigg] - \mathrm{KL}\big(q(\theta|\alpha)\|p(\theta)\big)\end{aligned} \tag{5}$$

where two Kullback-Leibler (KL) divergence terms are derived due to latent variables $\mathbf{z}$ and θ with the corresponding variational distributions $q_\eta(\mathbf{z}_n|\mathbf{x}_n)$ and $q(\theta|\alpha)$ and hyperparameters η and α, respectively. The optimization problem turns out to first marginalize the likelihood function over model parameters $\{\theta_g,\theta_d\}$ and then maximize the lower bound $\mathcal{L}(\eta,\alpha)$ with respect to the variational parameters $\{\eta,\alpha\}$ for generator and discriminator. It is noted that the first term in right hand side of Eq. (5) reflects the reconstruction loss. KL terms act as the regularization to match the variational distributions with the Gaussian prior densities in $p(\mathbf{z}_n)$ and $p(\theta)$ which are given by $\mathcal{N}(\mathbf{0},\mathbf{I})$. Marginalizing over latent variables characterizes

the randomness of latent structure in GAN which provides the capability to *outreach* different modes in data distribution. However, the log likelihood in GAN, i.e. the classification likelihood in big brackets in Eqs. (2)-(3), is intractable. This issue is tackled by replacing the likelihood $p_\theta(\mathbf{x}_n|\mathbf{z}_n)$ in variational lower bound $\mathcal{L}(\eta,\alpha)$ in Eq. (5) by using the synthesis likelihood $R(\theta)=\frac{p_\theta(\mathbf{x}_n|\mathbf{z}_n)}{p(\mathbf{x}_n)}\approx\frac{D(G(\mathbf{z}_n))}{1-D(G(\mathbf{z}_n))}$ [16] because two likelihoods are proportionally related. More specifically, the generator G with parameter θ_g is trained according to

$$\min_G \sum_{l=1}^{L_\theta}\Bigg\{\Bigg[\sum_{n=1}^{N}\Bigg\{-\mathbb{E}_{q_\eta(\mathbf{z}_n|\mathbf{x}_n)}\Bigg[\log\frac{D\big(G(\mathbf{z}_n,\theta_g^{(l)})\big)}{1-D\big(G(\mathbf{z}_n,\theta_g^{(l)})\big)}\Bigg] + \mathrm{KL}\big(q_\eta(\mathbf{z}_n|\mathbf{x}_n)\|p(\mathbf{z}_n)\big)\Bigg\}\Bigg] - \log q(\theta_g^{(l)}|\alpha_g)+\log p(\theta_g^{(l)})\Bigg\} \tag{6}$$

where the modified negative lower bound in a form of value function, expressed by $V_g(\eta,\alpha_g)\triangleq-\widetilde{\mathcal{L}}(\eta,\alpha_g)$, is minimized. The expectation over variational distribution $q(\theta_g|\alpha_g)$ is here approximated by L_θ Monte Carlo samples $\{\theta_g^{(l)}\}_{l=1}^{L_\theta}$. At the same time, the discriminator D with parameter θ_d is trained by maximizing the resulting variational function $V_d(\alpha_d)$ where the weight uncertainty is compensated according to

$$\max_D \sum_{l=1}^{L_\theta}\Bigg\{\sum_{n=1}^{N}\Bigg[\mathbb{E}_{\mathbf{x}_n\sim p(\mathbf{x}_n)}[\log D\big(\mathbf{x}_n,\theta_d^{(l)}\big)] + \mathbb{E}_{\mathbf{z}_n\sim p(\mathbf{z}_n)}[\log(1-D\big(G(\mathbf{z}_n\big),\theta_d^{(l)})]\Bigg] + \log q(\theta_d^{(l)}|\alpha_d)-\log p(\theta_d^{(l)})\Bigg\}\triangleq\max_D V_d(\alpha_d). \tag{7}$$

A minimax optimization is fulfilled to implement a two-player game for estimating G and D.

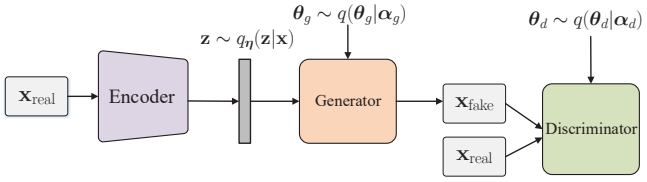

Fig. 1: Model structure for variational Bayesian GAN.

B. Implementation and algorithm

Figure 1 depicts the structure of the proposed variational GAN which consists of an encoder, a generator and a discriminator with the parameters or hyperparameters η, $\{\theta_g,\alpha_g\}$ and $\{\theta_d,\alpha_d\}$, respectively. The hybrid encoder and decoder in VGAN plays a similar role to those in VAE [17], [18]. The discriminator with hyperparameter α_d aims to distinguish the real sample $\mathbf{x}_{\text{real}}$ from the fake sample $\mathbf{x}_{\text{fake}}$ which is generated by reconstruction due to the encoder with parameter η and the decoder with hyperparameter α_g. Using VGAN, L_θ samples of parameters of generator and discriminator $\{\theta_g^{(l)},\theta_d^{(l)}\}$ are marginalized in the learning objective by using the variational distributions $\{q(\theta_g|\alpha_g),q(\theta_d|\alpha_d)\}$ which are driven by fully-connected neural networks (NNs) using parameters $\{\alpha_g,\alpha_d\}$. NN parameters $\{\eta,\alpha_g,\alpha_d\}$ are jointly estimated according to the learning procedure of VGAN as formulated in Algorithm 1. Three NNs $\{\eta,\alpha_g,\alpha_d\}$ are configured with the outputs of Gaussian mean and variance parameters $\{\mu_e,\rho_e,\mu_g,\rho_g,\mu_d,\rho_d\}$ which are used to draw samples $\{\mathbf{z}^{(l)},\theta_g^{(l)},\theta_d^{(l)}\}$ for implementation based on the stochastic gradient variational Bayes estimator [11]. The discriminator is updated k steps after the generator is updated once. The reparameterization trick for Gaussian sampling is applied to find stable samples θ_g and θ_d through the Gaussian parameters $\{\mu,\rho\}$ via sampling the standard Gaussian variable $\boldsymbol{\epsilon}$, i.e. $\theta=\mu+\log(1+\exp(\rho))\circ\epsilon$ where $\circ$ means element-wise product. This VGAN is realized by maximizing the variational lower bound of log marginal likelihood over the weights with twofold benefits. First, computational overhead in training VGAN is alleviated because the posterior sampling is avoided. Second, VGAN estimates a single set of parameters $\{\eta,\alpha_g,\alpha_d\}$ for data generation which significantly reduces the memory requirement than those for BGAN $\{\theta_g^{(l)},\theta_d^{(l)}\}_{l=1}^{L_\theta}$ where L_θ times of memory requirement is allocated.

Algorithm 1 Learning procedure for variational GAN

Require: $\eta=\{\mu_e,\rho_e\}$, $\alpha_g=\{\mu_g,\rho_g\}$, $\alpha_d=\{\mu_d,\rho_d\}$, L_θ, k
- **for** number of training iterations **do**
  - sample minibatch data from $p(\mathbf{x})$
  - **for** L_θ samples **do**
    - sample $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
    - calculate $\theta_g=\mu_g+\log(1+\exp(\rho_g))\circ\epsilon$
  - **end for**
  - update encoder η by minimizing $V_g(\eta,\alpha_g)$
  - update decoder α_g by minimizing $V_g(\eta,\alpha_g)$
  - **for** k steps **do**
    - sample minibatch data from $p(\mathbf{x})$
    - sample $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
    - calculate $\theta_d=\mu_d+\log(1+\exp(\rho_d))\circ\epsilon$
    - update discriminator α_d by maximizing $V_d(\alpha_d)$
  - **end for**
- **end for**

IV. Experiments

Different GAN models were evaluated by using the synthesized samples over two learning tasks.

A. MNIST task

The Mixed National Institute of Standards and Technology (MNIST) database consisted of handwritten digits from 0 to 9 and was widely used for evaluation of different learning tasks. This database contained 60,000 training images and 10,000 test images. Each digit was centered in a gray-scale image of size 28×28. This dataset was used to conduct evaluation for regression as well as classification. In regression task, we estimated the generative models based on VAE [11], GAN [1], Bayesian GAN [13] and the proposed variational GAN for comparison. There were three hidden layers in neural networks for encoder, generator and discriminator and $L_\theta=5$. Using

BGAN, 10 sets of generator weights were assumed and sampled. Adam optimization was used. In classification task, we compare the results of BGAN and VGAN based on the semi-supervised learning where the model structure was arranged to make label predictions [19], [20]. Both unlabeled and labeled data were used. Discriminator was not only used to distinguish fake data from real data but also classify the sample into one of 10 classes. The setting of semi-supervised learning provided a practical approach to desirable classification performance using GANs with the limited amount of labeled data.

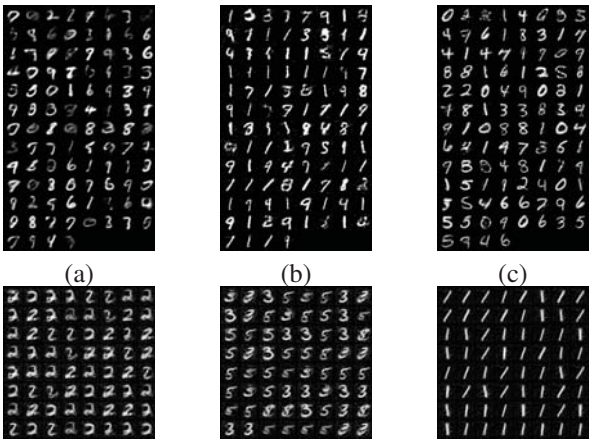

Fig. 2: MNIST digits from random noise by using (a) VAE, (b) GAN, (c) variational GAN, and (d)-(f) Bayesian GAN using three different samples of generator weights $\theta_g^{(l)}$.

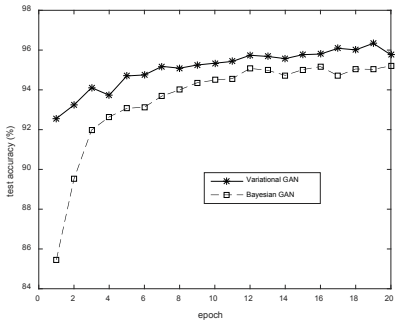

Fig. 3: Classification accuracy of test data versus learning epochs using Bayesian GAN and variational GAN.

Figure 2 displays the random samples of MNIST digits by using VAE, GAN, BGAN and VGAN. A regression task with unsupervised learning is performed. The samples generated by VAE look blurred while those generated by GAN look precisely. However, the mode collapse happens because the mode of digit '1' is frequently drawn. But, several other digits or modes are rarely observed. Using Bayesian GAN, we show the generated images based on three selected parameter samples (totally 10 parameter sets of $\theta_g^{(l)}$ are required). It is obvious that BGAN does obtain meaningful samples of generator weights so that three sets of generated samples sufficiently reflect three different modes. Nevertheless, the proposed variational GAN (in Figure 2(c)) produces images with good quality and diversity based on *a single parameter* α_g. θ_g has been marginalized. Basically, different modes in BGAN produce the meaningful samples with similar classes. However, BGAN requires multiple parameter sets for Bayesian integration. Figure 3 illustrates the classification accuracy of test data versus the number of learning epochs by using BGAN and VGAN. The more learning epochs the models are trained by BGAN and VGAN, the higher the classification performance is achieved. Obviously, the convergence of learning procedure using VGAN is faster than that using BGAN. Accuracies using VGAN are higher than those using BGAN.

	BGAN	VGAN
generator	35748	14392
discriminator	2764	213
total	38512	14605

TABLE I: Comparison of number of parameters in GANs.

B. CelebA task

CelebA [21] is known as a large-scale task for face attributes with more than 200K celebrity images, each with 40 attributes. The color images in this dataset covered large pose variations and background clutters. CelebA had a large variety of images with rich annotations where there were 10,177 identities and 202,599 face images from 5 landmark locations with 40 binary attributes per image. We ran the experiment on image generation by using the aligned images which were cropped by using the similarity transformation according to two eye locations. In practice, we resized each image to a fixed dimension of 64x64 ($\mathbf{x} \in \mathbb{R}^{64x64x3}$) and trained without attributes. In such an unsupervised learning, the convolutional layers were consistently used in GAN (also known as the deep convolutional GAN [22]), BGAN and VGAN. The encoder and decoder used the kernel size of 5 and 4, respectively. There were four convolutional layers and one fully-collected layer in the encoder with dimensions 64, 128, 256, 512, 64 and one fully-connected and four convolutional layers in the decoder with dimensions 512, 256, 128, 64, 32, respectively. Softplus activation was used with batch normalization for both encoder and decoder. The discriminator was configured as three fully-connected layers with the same dimension 512. Latent code had dimension $\mathbf{z} \in \mathbb{R}^{64}$. ReLU was used. Table I compares the size of parameters in the generator and discriminator using BGAN and VGAN. Both GANs adopt the Bayesian convolutional layers and fully-connected layers. BGAN uses much more parameters than VGAN.

Figure 4 displays a number of samples generated by BGAN. The synthesized images look clearly as the human faces with reasonable shape in face content, hair and background. But, the eyes, noses, mouses, face skins, hairs are partially broken, twisted or non-smoothed. By using VGAN, the synthesized images are improved. To quantify the quality of the generated face images, we sample 1K and 10K of synthesized Celeb

(a) (b)

Fig. 4: CelebA faces by using (a) BGAN and (b) VGAN.

faces and measure the Fréchet inception distance (FID) [23] which compares the statistics or calculates the similarity between the generated samples to real images. FID was shown to characterize the disturbance of generated images [24]. The Fréchet distance between two multivariate Gaussians of real samples and generated samples with means and covarances $\{\mu_r, \Sigma_r\}$ and $\{\mu_g, \Sigma_g\}$, respectively, is calculated by

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}). \quad (8)$$

This FID is measured from the Gaussians of real and generated samples which are calculated from the 2048-dimensional activations in the last layer of generator. The smaller the FID, the better the generated images are obtained. Table II reports the FID values by using different GANs. 10K samples obtains smaller FID than 1K samples. The proposed VGAN achieves the lowest FID when compared with GAN and BGAN.

	GAN	BGAN	VGAN
1K	107.2	70.4	22.5
10K	95.8	60.7	10.3

TABLE II: Comparison of FIDs by using different GANs.

V. Conclusion

This paper explored the Bayesian learning for generative adversarial networks which tackled the issue of mode collapse by regularizing the trained models where the weight uncertainty was compensated. The learning objective was derived to implement the variational generative adversarial network via a minimax optimization for estimation of variational parameters for encoder, generator and discriminator where the prior distributions were merged. Mode collapse was mitigated. Theoretical illustration and comparison with other generative models were addressed. The variational Bayesian variant of GAN was proposed with desirable performance in a regression task for unsupervised learning as well as a classification task for semi-supervised learning as shown in the experiments. Two learning tasks on real-world data with different sizes were examined over different generative models to show the merit of the proposed method in terms of visualization and Fréchet inception distance in the learning procedure. Future works will be extended to implement variational GAN for generation of other types of technical data.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[2] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[3] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[4] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *arXiv preprint ArXiv:1611.04076*, 2016.

[5] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed, "Variational approaches for auto-encoding generative adversarial networks," *arXiv preprint arXiv:1706.04987*, 2017.

[6] J.-C. Tsai and J.-T. Chien, "Adversarial domain separation and adaptation," in *Prof. of IEEE International Workshop on Machine Learning for Signal Processing*, 2017, pp. 1–6.

[7] J.-T. Chien and K.-T. Peng, "Adversarial manifold learning for speaker recognition," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, 2017, pp. 599–605.

[8] J.-T. Chien and K.-T. Peng, "Adversarial learning and augmentation for speaker recognition," in *Proc. of Speaker and Language Recognition Workshop*, 2018, pp. 342–348.

[9] J.-T. Chien and Y.-Y. Lyu, "Partially adversarial learning and adaptation," in *Proc. of European Signal Processing Conference*, 2019.

[10] J.-T. Chien and K.-T. Peng, "Neural adversarial learning for speaker recognition," *Computer Speech and Language*, 2019.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representations*, 2014.

[12] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network for language modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 361–374, 2016.

[13] Y. Saatci and A. G. Wilson, "Bayesian GAN," in *Advances in Neural Information Processing Systems*, 2017, pp. 3622–3631.

[14] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arXiv:1512.09300*, 2015.

[15] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[16] R. Dutta, J. Corander, S. Kaski, and M. U. Gutmann, "Likelihood-free inference by ratio estimation," *arXiv preprint arXiv:1611.10242*, 2016.

[17] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[18] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," in *Proc. of International Conference on Learning Representations*, 2017.

[19] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[20] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," *arXiv preprint arXiv:1610.09585*, 2016.

[21] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. of International Conference on Computer Vision*, 2015.

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

[23] D. C. Dowson and B. V. Landau, "The Fréchet distance between multivariate normal distributions," *Journal of Multivariate Analysis*, vol. 12, no. 3, pp. 450–455, 1982.

[24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.