AE-OT-GAN: Training GANs from data specific latent distribution

Dongsheng An[1], Yang Guo[1], Min Zhang[2], Xin Qi[1], Na Lei[⋆3], and Xianfang Gu[1]

[1] Stony Brook University
[2] Harvard Medical School
[3] Dalian University of Technology
{doan, yangguo, xinqi, gu}@cs.stonybrook.edu, mzhang@bwh.harvard.edu

Abstract. Though generative adversarial networks (GANs) are prominent models to generate realistic and crisp images, they are unstable to train and suffer from the mode collapse problem. The problems of GANs come from approximating the intrinsic discontinuous distribution transform map with continuous DNNs. The recently proposed AE-OT model addresses the discontinuity problem by explicitly computing the discontinuous optimal transform map in the latent space of the autoencoder. Though have no mode collapse, the generated images by AE-OT are blurry. In this paper, we propose the AE-OT-GAN model to utilize the advantages of the both models: generate high quality images and at the same time overcome the mode collapse problems. Specifically, we firstly embed the low dimensional image manifold into the latent space by autoencoder (AE). Then the extended semi-discrete optimal transport (SDOT) map is used to generate new latent codes. Finally, our GAN model is trained to generate high quality images from the latent distribution induced by the extended SDOT map. The distribution transform map from this dataset related latent distribution to the data distribution will be continuous, and thus can be well approximated by the continuous DNNs. Additionally, the paired data between the latent codes and the real images gives us further restriction about the generator and stabilizes the training process. Experiments on simple MNIST dataset and complex datasets like CIFAR10 and CelebA show the advantages of the proposed method.

Keywords: Generative Model, Optimal Transport, GAN, Continuity

1 Introduction

Image generation has been one of the core topics in the area of computer vision for a long time. Thanks to the quick development of deep learning, numerous generative models are proposed, including encoder-decoder based models [2, 20, 44], generative adversarial networks (GANs) [3, 6, 14, 15, 37, 47], density estimator based models [8, 9, 21, 35] and energy based models [24, 34, 48, 51]. The encoder-decoder based models and GANs are the most prominent ones due to their capability to generate high quality images.

Intrinsically, the generator in a generative model aims to learn the real data distribution supported on the data manifold [43]. Suppose the distribution of a specific class of natural data ν_{gt} is concentrated on a low dimensional manifold χ embedded in the high

⋆ Corresponding author: nalei@dlut.edu.cn

dimensional data space. The encoder-decoder methods first attempt to embed the data into the latent space Ω through the encoder f_θ, then samples from the latent distribution are mapped back to the manifold to generate new data by decoder g_ξ. While GANs, which have no encoder, directly learn a map (generator) that transports a given prior low dimensional distribution to ν_{gt}.

Usually, GANs are unstable to train and suffer from mode collapse [13, 30]. The difficulties come from the fact that the generator of a GAN model is trained to approximate the discontinuous distribution transport map from the *unimodal Gaussian distribution* to the *real data distribution* by the continuous neural networks [2, 19, 47]. In fact, when the supporting manifolds of the source and target distributions differ in topology or convexity, the OT map between them will be discontinuous [45]. Distribution transport maps can have complicated singularities, even when the ambient dimension is low [12]. This poses a great challenge for the generator training in standard GAN models.

To tackle the mode collapse problem caused by discontinuous transport maps, the authors of [2] proposed the AE-OT model. In this model, an autoencoder is used to map the image manifold χ into the latent manifold Ω. Then, the semi-discrete optimal transport (SDOT) map T from the uniform distribution $Uni([0,1]^d)$ to the empirical latent distribution is explicitly computed via convex optimization approach. Then a piece-wise linear extension map of the SDOT, denoted by $\tilde{T}$, pushes forward the uniform distribution to a continuous latent distribution μ, which in turn gives a good approximation of the latent distribution $\mu_{gt} = f_{\theta\#}\nu_{gt}$ ($f_{\theta\#}$ means the push forward map induced by f_θ). Composing the continuous decoder g_ξ and discontinuous $\tilde{T}$ together, i.e. $g_\xi \circ \tilde{T}(w)$, where w is sampled from uniform distribution, this model can generate new images. Though have no mode collapse, the generated images look blurry.

In this work we propose the AE-OT-GAN framework to combine the advantages of the both models and generate high quality images without mode collapse. Specifically, after the training of the autoencoder and the computation of the extended SDOT map, we can directly sample from the latent distribution μ by applying $\tilde{T}(w)$ on the uniform distribution to train the GAN model. In contrast to the conventional GAN models, whose generators are trained to transport the latent Gaussian distribution to the data manifold distribution, our GAN model sample from the data inferred latent distribution μ. The distribution transport map from μ to the data distribution ν_{gt} is continuous and thus can be well approximated by the generator (parameterized by CNNs). Moreover, the decoder of the pre-trained autoencoder gives a warm start of the generator, so that the Kullback–Leibler divergence can be directly applied in the discriminator because the real and fake batches of images have non-vanishing overlap in their supports during the training phase. Furthermore, the content loss and feature loss between the paired latent codes and real input images regularize the adversarial loss, stabilize the GAN training and help get rid of mode collapse problem. Experiments have shown efficacy and efficiency of our proposed model.

The contributions of the current work can be summarized as follows: **(1)** This paper proposes a novel AE-OT-GAN model that combines the strengths of AE-OT model and GAN model. The proposed model removes the blurriness of the images generated by AE-OT, and at the same time keep the good properties of the latter in eliminating the mode collapse problems. **(2)** The decoder of the autoencoder provides a good initialization of

the generator of GAN, which makes the supports of the real and fake image distributions overlap and thus the KL divergence can be used in the discriminator. **(3)** In addition to the adversarial loss, the explicit correspondence between the latent codes and the real images provide auxiliary constraints, namely the content loss and feature loss, to the generator. The both losses make sure that there is no mode collapse in our model. **(4)** The experiments demonstrate that our model can generate images consistently better than the results of state-of-the-art methods.

2 Related Work

The proposed method in this paper is highly related to encoder-decoder based generation models, the generative adversarial networks (GANs), conditional GANs and the hybrid models that take the advantages of above.

Encoder-decoder architecture A breakthrough for image generating comes from the scheme of Variational Autoencoders (VAEs) (e.g. [20]), where the decoders approximate real data distributions from a Gaussian distribution in a variational approach (e.g [20] and [39]). Latter Yuri Burda et al. [5] lower the requirement of latent distribution and propose the importance weighted autoencoder (IWAE) model through a different lower bound. Bin and David [7] propose that the latent distribution of VAE may not be Gaussian and improve it by firstly training the original model and then generating new latent code through the extended ancestral process. Another improvement of the VAE is the VQ-VAE model [36], which requires the encoder to output discrete latent codes by vector quantisation, then the posterior collapse of VAEs can be overcome. By multi-scale hierarchical organization, this idea is further used to generate high quality images in VQ-VAE-2 [38]. In [44], the authors adopt the Wasserstein distance in the latent space to measure the distance between the distribution of the latent code and the given one and generate images with better quality. Different from the the VAEs, the AE-OT model [2] firstly embed the images into the latent space by autoencoder, then an extended semi-discrete OT map is computed to generate new latent code based on the fixed ones. Decoded by the decoder, new images can be generated. Although the encoder-decoder based methods are relatively simple to train, the generated images tend to be blurry.

Generative adversarial networks The GAN model [14] tries to alternatively update the generator, which maps the noise sampled from a given distribution to real images, and the discriminator differentiates between the generated images and the real ones. If the generated images successfully fool the discriminator, the model is well trained. Later, [37] proposes a deep convolutions neural network (DCGAN) to generate images with better quality. While being a powerful tool in generating realistic samples, GANs can be hard to train and suffer from mode collapse problem [13]. After delicate analysis, [3] points out that it is the KL divergence the original GAN used causes these problems. Then the authors introduced the celebrated WGAN, which makes the whole framework easy to converge. To satisfy the Lipschitz continuity required by WGAN, a lot of methods are proposed, including clipping [3], gradient penalty [15], spectral normalization [33] and so on. Later, Wu et al. [46] use the Wasserstein divergence objective, which get rid of the Lipschitz approximation problem and get a better result. Differently, the OT-GAN [41] uses the Sinkhorn algorithm to approximate the Wasserstein distance in the image space. Instead of L_1 cost adopted by WGAN, Liu et.al [29] propose the WGAN-QC by taking

the L_2 cost into consideration. Though various GANs can generate sharp images, they will theoretically encounter the mode collapse problem [2, 13].

Hybrid models To solve the blurry image problem of encoder-decoder architecture and the mode collapse problems of GANs, a natural idea is to compose them together. Larsen et al. [23] propose to combine the variational autoencoder with a generative adversarial network, and thus generate images better than VAEs. [32] matches the aggregated posterior of the hidden code vector of the autoencoder with an arbitrary prior distribution by a discriminator and then applies the model into tasks like semi-supervised classification and dimensionality reduction. BiGAN [10], with the same architecture with ours, uses the discriminator to differentiate both the generated images and the generated latent code. Further, utilizing the BigGAN generator [4], the BigBiGAN [11] extends this method to generate much better results. Here we also treat the BourGAN [47] as a hybrid model, because it firstly embeds the images into latent space by Bourgain theorem, then trains the GAN model by sampling from the latent space using the GMM model.

Conditional GANs are another kind of hybrid models that can also be treated as image-to-image transformation. For example, using an encoder-decoder architecture to build the connection between paired images and then differentiating the decoded images with the real ones by a discriminator, [17] is able to transform images of different styles. Further, SRGAN [26] uses similar architecture to get super resolution images from their low resolution versions. The SRGAN model is the most similar work to ours, as it also utilizes the content loss and adversarial loss. The main differences between this model and ours including: (i) SRGAN just uses the paired data, while the proposed method use both the paired data and generated new latent code to train the model; (ii) the visually meaningful features used by SRGAN are extracted from the pre-trained VGG19 network [42], while in our model, they come from the encoder itself. This makes them more reasonable especially under the scenes where the datasets are not included in those used to train the VGG.

Note that no mode collapse in the AE-OT model cannot directly guarantee that there is no mode collapse of the AE-OT-GAN model. For the AE-OT model, the pre-trained decoder of the AE is used as generator, thus if there is no mode collapse in the latent space, there will be no mode collapse in the image space. For the AE-OT-GAN model, the decoder is changed. The elimination of the mode collapse is thus guaranteed by the paired content loss between the latent codes and the real images.

3 The Proposed Method

In this section, we explain our proposed AE-OT-GAN model in detail. There are mainly three modules, an autoencoder (AE), an optimal transport mapper (OT) and a GAN model. Firstly, an AE model is trained to embed the data manifold χ into the latent space. At the same time, the encoder f_θ pushes forward the ground-truth data distribution ν_{gt} supported on χ to the ground-truth latent distribution μ_{gt} supported on Ω in the latent space. Secondly, we compute the semi-discrete OT map from the uniform distribution to the discrete empirical latent distribution $\hat{\mu}_{gt}$. By the extended SDOT map $\tilde{T}$, we can construct the continuous distribution μ that approximates the ground-truth latent distribution μ_{gt} well. Finally, starting from μ as the latent distribution, our GAN model is trained to generate both realistic and crisp images. The pipeline of our proposed model is illustrated in Fig. 1. In the following, we will explain the three modules one by one.

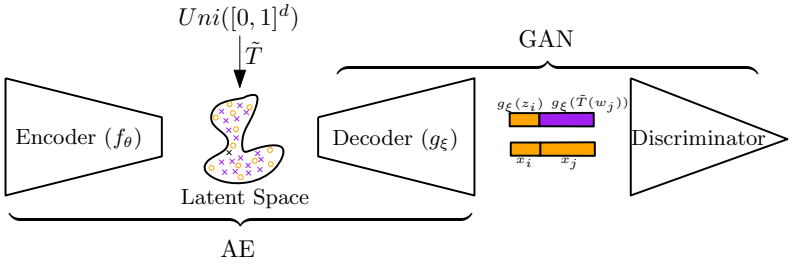


Fig. 1. The framework of the proposed method. Firstly, the autoencoder is trained to embed the images into the latent space, the real latent codes are shown as the orange circles. Then we compute the extended semi-discrete OT map $\tilde{T}$ to generate new latent codes in the latent space (the purple crosses). Finally, our GAN model is trained from the latent distribution $\mu = \tilde{T}_{\#}Uni([0,1]^d)$ to the image distribution. Here the generator is just the decoder of the autoencoder. The fake batch (the bar with orange and purple colors) to train the discriminator is composed of two parts: the reconstructed images $g_\xi(z_i)$ of the real latent codes and the generated images $g_\xi(\tilde{T}(w))$ from the randomly generated latent codes with w sampled from uniform distribution. The real batch (the bar with only orange color) is also composed of two parts: the real images x_i corresponding to z_i, and the randomly selected images x_j.

3.1 Data Embedding with Autoencoder

We model the real data distribution as a probability measure ν_{gt} supported on an r dimensional manifold χ embedded in the D dimensional Euclidean space $\mathbb{R}^D$ (ambient space) with $r \ll D$. In the first step of our AE-OT-GAN model, we train an autoencoder (AE) to embed the real data manifold χ to be the latent manifold Ω. In particular, training the AE model is equivalent to compute the encoding map f_θ and decoding map g_ξ

$$(\nu_{gt}, \chi) \xrightarrow{f_\theta} (\mu_{gt}, \Omega) \xrightarrow{g_\xi} (\nu_{gt}, \chi)$$

by minimizing the loss function:

$$\mathcal{L}(\theta, \xi) := \sum_{i=1}^{n} \|x_i - g_\xi \circ f_\theta(x_i)\|^2,$$

with f_θ and g_ξ parameterized by standard CNNs (θ and ξ are the parameters of the networks, respectively). Given a dense sampling from the image manifold (detailed explanation is included in the supplementary) and ideal optimization (namely the loss function goes to 0), $f_\theta \circ g_\xi$ coincides with the identity map. After training, f_θ is a continuous, convertible map, namely a *homeomorphism*, and g_ξ is the inverse homeomorphism. This means $f_\theta : \chi \to \Omega$ is an embedding, and pushes forward ν_{gt} to the latent data distribution $\mu_{gt} := f_{\theta\#}\nu_{gt}$. In practice, we only have the empirical data distribution given by $\hat{\nu}_{gt} = \frac{1}{n}\sum_{i=1}^{n} \delta(x - x_i)$, which is pushed forward to be the empirical latent distribution $\hat{\mu}_{gt} = \frac{1}{n}\sum_{i=1}^{n} \delta(z - z_i)$, where n is the number of samples.

3.2 Constructing μ with Semi-Discrete OT Map

In this section, from the empirical latent distribution $\hat{\mu}_{gt}$, we construct a continuous latent distribution μ following [2] such that (i) it generalizes $\hat{\mu}_{gt}$ well, so that all of the

modes in the latent space are covered by the support of μ (ii) the support of μ has similar topology to that of μ_{gt}, which ensures that the transport map from μ to ν_{gt} has less discontinuities and (iii) it is efficient to sample from μ.

To obtain μ, the semi-discrete OT map T from the uniform distribution $Uni([0,1]^d)$ to the empirical latent distribution $\hat{\mu}_{gt}$ is firstly computed. Here d is the dimension of the latent space. By extending T to be a piece-wise linear map $\tilde{T}$, we can construct μ as the push forward distribution of $Uni([0,1]^d)$ under $\tilde{T}$:

$$(Uni([0,1]^d), [0,1]^d) \xrightarrow{\tilde{T}} (\mu, \Omega)$$

Theorem 1. *The 2-Wasserstein distance between μ and $\hat{\mu}_{gt}$ satisfies $W_2(\mu, \hat{\mu}_{gt}) \leq \varepsilon$, where ε is a given constant to build μ. Moreover, if the latent codes are densely sampled from the latent manifold Ω, we have $W_2(\mu, \mu_{gt}) \leq 2\varepsilon$, μ-almost surely.*

The construction details of μ can be found in [2] and the supplementary, and we also give the proof of the above theorem in the supplementary. This theorem tells us that as a continuous generalization of $\hat{\mu}_{gt}$, μ is a good approximation of μ_{gt}. Also, we want to mention that $\tilde{T}$ is a piece-wise linear map that pushes forward $Uni([0,1]^d)$ to μ, which makes the sampling from μ efficient and accurate. Based on the construction of $\tilde{T}$, the sampling from μ is equivalent to the locally piece-wise linear interpolation of z_is in the latent space, which guarantees that there is no mode collapse in μ.

3.3 GAN Training from μ

The GAN model computes the transport map from the continuous latent distribution μ to the data distribution on the manifold.

$$(\mu, \Omega) \xrightarrow{g_\xi} (\nu_{gt}, \chi).$$

Our GAN model is based on the vanilla GAN model proposed by Ian Goodfellow et.al [14]. The generator g_ξ is used to generate new images by sampling from the latent distributin μ, while the discriminator d_η is used to discriminate if the distribution of the generated images are the same with that of the real images. The training process is formalized to be a min-max optimization problem:

$$\min_\xi \max_\eta \mathcal{L}(\xi, \eta),$$

where the loss function is given by

$$\mathcal{L}(\xi, \eta) = \mathcal{L}_{adv} + \mathcal{L}_{feat} + \beta \mathcal{L}_{img} \tag{1}$$

In our model, the loss function consists of three terms, the adversarial loss $\mathcal{L}_{adv}$, the image content loss $\mathcal{L}_{img}$ and the feature loss $\mathcal{L}_{feat}$. Here $\beta > 0$ is the weight of the content loss.

Adversarial Loss We adopt the vanilla GAN model [14] based on the Kullback–Leibler (KL) divergence. The key difference between our model and the original GAN is that

our latent samples are drawn from the data related latent distribution μ, instead of the Gaussian distribution. The adversarial loss is given by:

$$\mathcal{L}_{adv} = \min_{\xi} \max_{\zeta} E_{x\sim\nu_{gt}}[log\ d_\zeta(x)] + E_{z\sim\mu}[log(1 - d_\zeta(g_\xi(z)))]$$

According to [3], vanilla GAN is hard to converge because the supports of the distributions of the real images and fake images may not intersect each other, which makes the KL divergence between them infinity. This issue is solved in our case, because (1) the training of AE gives a warm start to the generator, so at the beginning of the training, the support of the generated distribution $g_{\xi\#}\mu$ is close to that of the real data distribution ν_{gt}; (2) by delicate settings of the fake and real batches used to train the discriminator, we can keep the KL divergence between them converge well. In detail, as shown in Fig. 1, the fake batch is composed of both the reconstructed images from the real latent codes (the orange circles) and the generated images from the generated latent codes (the purple crosses), and the real batch includes both the real images corresponding to the real latent codes and some randomly selected real images.

Content Loss Recall that the generator can produce two types of images: images reconstructed by real latent codes and images from generated latent codes. Given a real sample x_i, its latent code is $z_i = f_\theta(x_i)$, the reconstructed image is $g_\xi(z_i)$. Each reconstructed image is represented as a triple $(x_i, z_i, g_\xi(z_i))$. Suppose there are n reconstructed images in total, the content loss is given by

$$\mathcal{L}_{img} = \frac{1}{n}\sum_{i=1}^{n} \|g_\xi(z_i) - x_i\|_2^2 \tag{2}$$

Where g_ξ is the generator parameterized by ξ.

Feature Loss We adopt the feature loss similar to that in [26]. Given a reconstructed image triple $(x_i, z_i, g_\xi(z_i))$, we encode $g_\xi(z_i)$ by the encoder of AE. Ideally, the real image x_i and the generated image $g_\xi(z_i)$ should be the same, therefore their latent codes should be similar. We measure the difference between their latent codes by the feature loss. Furthermore, we can measure the difference between their intermediate features from different layers of the encoder.

Suppose the encoder is a network with L layers, the output of the lth layer is denoted as $f_\theta^{(l)}$. The feature loss is given by

$$\mathcal{L}_{feat} := \frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{L} \alpha^{(l)} \| f_\theta^{(l)}(x_i) - f_\theta^{(l)} \circ g_\xi(z_i)\|_2^2,$$

Where $\alpha^{(l)}$ is the weight of the feature loss of the l-th layer.

For reconstructed images $(x_i, z_i, g_\xi(z_i))$, the content loss and the feature loss force the generated image $g_\xi(z_i)$ to be the same with the real image x_i. Therefore the eliminating of mode collapse in the latent space means that there is no mode collapse in the image space.

3.4 Geometric perspective of AE-OT-GAN

Another perspective of the proposed model is that it can be treated as a manifold fitting framework. Ideally, if given an embedding map $f : \chi \to \Omega$ and a dense dataset X

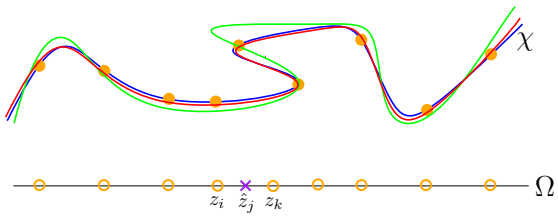


Fig. 2. Manifold fitting result of the decoder/GAN. The blue curve is the original manifold. The green one shows the fitting result of the AE-OT model. By the AE-OT-GAN framework, we can not only draw $g_\xi(z_i)$ much closer to x_i, the whole manifold (the red curve) also fit the original one (blue curve) better. The orange circles on Ω represent the real latent codes, and the purple one represents the generated latent code. The orange disks on the manifold represent real data.

sampled from a distribution ν_{gt} supported on χ, the purpose of the generation model is to generate new samples following the distribution of ν_{gt} and locating on the manifold χ. For the AE-OT model [2], it only requires that the reconstructed images should be similar to the real ones under L_2 distance. As a result, the support of the generated image distribution may only fit the real manifold χ well near the given samples. As shown in Fig. 2, the orange circles represent the latent codes, and the green curve represents the support of the generated distribution of AE-OT model, which only fits the real manifold χ well nearby the given samples. For the AE-OT-GAN model, on one hand, the feature loss and content loss require that the reconstructed manifold (the red curve of Fig. 2) should approach to the real manifold χ on the given samples; on the other hand, the discriminator is used to regularize the fitting performance of the generated manifold on both the given samples and new generated samples, namely both the reconstructed images $g_\xi(z_i)$ and the generated images $g_\xi(\hat{z}_j)$ should fit the real manifold well. Here z_i and $\hat{z}_j$ represent the real latent codes and the generated latent codes. Therefore, the generated manifold by the AE-OT-GAN model fits the real manifold χ far more better than the AE-OT model. Moreover, according to Sec. 3.2, generating a new latent code from μ is essentially equivalent to locally linear interpolation by the real latent codes. As a result, the generated images can actually be treated as the non-linear interpolation by the nearby real images. For example, $\hat{z}_j$ is generated by linear interpolation between z_i and z_k, then the location of $g_\xi(z_i)$ should be between x_i and x_k.

4 Experiments

To evaluate the proposed method, experiments are conducted on various datasets including MNIST [25], stack MNIST [28], Cifar10 [22], CelebA [50] and CelebA-HQ [27].

Evaluation metrics To illustrate the performance of the proposed method, we adopt the commonly used Frechet Inception distance (FID) [16] as our main evaluation metrics. When the images are embedded into the feature space by inception network, two high dimensional Gaussian distributions are used to approximate the empirical distributions of the generated and real features, respectively. The FID is given by the difference between the two Gaussian distributions. Lower FID means better quality of the generated dataset. For the Cifar10 dataset, another popular metric is the Inception Score (IS) [40], which

can be used to measure the quality of each single image. Higher IS means better quality of the generated image.

Training details To get rid of the vanishing gradient problem and make the model converge better, we use the following three strategies:

(i) Train the discriminator using Batch Composition There are two types of latent codes in our method: *the real latent codes* coming from encoding the real images by the encoder, and the generated latent codes coming from the extended SDOT map. Correspondingly, there are two types of generated images, *the reconstructed images* from the real latent codes and *the generated images* from the generated latent codes.

To train the discriminator, both the fake batch and real batch are used. *The fake batch* consists of both randomly selected reconstructed images and generated images, and *the real batch* only includes real images, in which the first part has a one-to-one correspondence with the reconstructed images in the fake batch, as shown in Fig. 1. In all the experiments, the ratio between the number of generated images and reconstructed images in the fake batch is 3. This strategy ensures that there is an overlap between the supports of the fake and real batches, so that the KL divergence is not infinity.

(ii) Different learning rate For better training, we use different learning rates for the generator and the discriminator as suggested by Heusel et al. in [16]. Specifically, we set the learning rate of the generator to be $lr_G = 2e - 5$ and that of the discriminator to be $lr_D = lr_G/R$, where $R > 1$. This improves the stability of the training process.

(iii) Different inner steps Another way to improve the training consistency of the whole framework is to set different update steps for the generator and discriminator. Namely, when the discriminator updates once, the generator updates S times correspondingly. This strategy is opposite to the training of vanilla GANs, which typically require multiple discriminator update steps per generator update step.

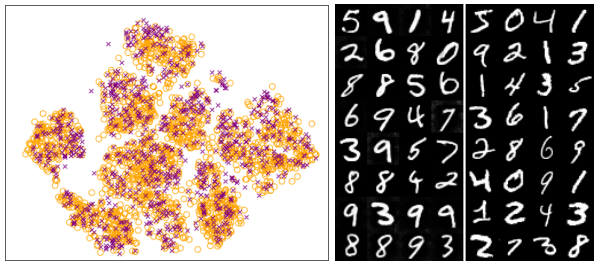
By setting R and S, we can keep the discriminator output of the real images slightly large than that of the generated ones, which can better guide the training of the generator. For the MNIST and stack MNIST datasets, $R = 15$ and $S = 3$; for the Cifar10 dataset, $R = 25$ and $S = 10$; and for the CelebA and CelebA-HQ datasets, $R = 15$ and $S = 5$. In Eq. 1, $\beta = 2000$ and $\alpha^{(l)} = 0.06$ with $l < L$, where L denotes the last layer of the encoder. $\alpha^L = 2.0/\|Z\|_2$ is used to regularize the loss of the latent codes.

With the above settings and the warm initialization of the generator from the pre-trained decoder, for each dataset, the total epochs will be less than 1000.

4.1 Convergence Analysis on MNIST

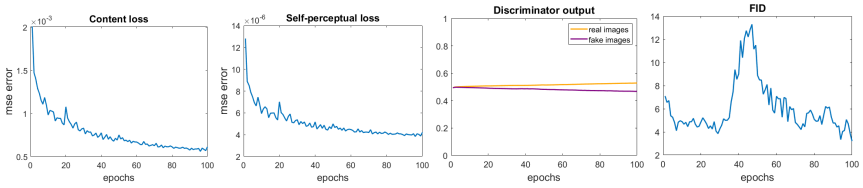
In this experiment, we evaluate the performance of our proposed model on MNIST dataset [25], which can be well embedded into the 64 dimensional latent space with the architecture of InfoGAN [6]. In Fig. 3(a), we visualize the real latent codes (orange circles) and the generated latent codes (purple crosses) by t-SNE [31]. It is obvious that the support of the real latent distribution and that of the generated latent distribution align well. Frame (b) of Fig. 3 shows the comparison between the generated handwritten digits (left) and the real digits (right), which is very difficult for humans to distinguish.

To show the convergent property of the proposed method, we plot the related curves in Fig. 4. The frame (a) and (b) show the changes of the content loss and the feature loss, and both of them decrease monotonously. The frame (c) shows that the output of the



(a) (b)

Fig. 3. (a) Latent code distribution. The orange circles represent the fixed latent code and the purple crosses are the generated ones. (b) Comparison between the generated digits (left) and the real digits (right).



(a) (b) (c) (d)

Fig. 4. The curves for training on MNIST dataset [25] of each epoch, including the results of content loss (a) and self-perceptual loss (b), the discriminator output (c) and FIDs (d).

discriminator for real images is only slightly larger than that for the fake images during the training process, which can help the generator generate more realistic digits. The frame (d) gives the evolution of FID and the final value is 3.2. For MNIST dataset, the best known FIDs with the same InfoGAN architecture are 6.7 and 6.4, reported in [30] and [2] respectively. This shows our model outperforms the state-of-the-art.

4.2 Mode Collapse Analysis on Stack MNIST

In this section, we test the diversity of the generated samples for the proposed AE-OT-GAN model on stack MNIST dataset [28], which includes 1,000 modes in total. The AE module of the AE-OT-GAN is consistent with [2] and the architecture of the discriminator is set to be the same as the encoder with the final output to be a scalar. The number of modes and the reverse KL divergence are used as the metrics to test the mode collapse performance. In Tab. 1, we show the results of the proposed method and the comparisons including DCGAN [37], VEEGAN [1], PacGAN [28], WGAN [3] and AE-OT [2]. It is obvious that the AE-OT-GAN model keeps the 'no-mode-collapse' property of the AE-OT model and has no mode miss in the generated images.

4.3 Quality Evaluation on Complex Dataset

In this section, we compare with the SOTA methods both quantitatively and qualitatively. The standard and ResNet models used to train the Cifar10 dataset are the same with those used by SNGAN [33], and the architectures of WGAN-div [46] are used to train the CelebA dataset. The architecture used to train the CelebA-HQ dataset is illustrated

Table 1. Experiments on stacked MNIST.

	Stacked MNIST	
	Modes	KL
DCGAN	99.0	3.40
VEEGAN	150.0	2.95
PacDCGAN4	1000.0 ± 0.00	0.07 ± 0.005
WGAN(*)	314.3 ± 38.54	2.44 ± 0.170
AE-OT(*)	1000.0 ± 0.00	0.03 ± 0.0008
AE-OT-GAN(*)	1000.0 ± 0.0	0.05 ± 0.006

(a) Epoch 0 (b) Epoch 80 (c) Epoch 160 (d) Epoch 240 (e) Ground-truth

Fig. 5. Evolution of the generator during training on the CelebA dataset [50].

in the supplementary. The frameworks of the encoders are just set to be the mirror of the corresponding generators/decoders. **Progressive Quality Improvement** Firstly, we

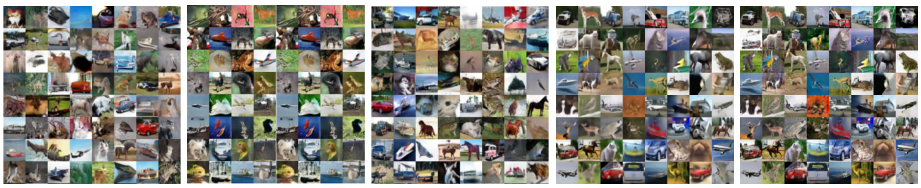
	CIFAR10				CelebA	
	Standard		Resnet		Standard	Resnet
	FID	IS	FID	IS	FID	FID
WGAN-GP [15]	40.2	6.68	19.6	7.86	21.2	18.4
PGGAN [18]	-	-	18.8	8.80	-	16.3
SNGAN [33]	25.5	7.58	21.7	8.22	-	-
WGAN-div [46]	-	-	18.1	-	17.5	15.2
WGAN-QC [29]	-	-	-	-	-	12.9
AE-OT [2]	34.2	6.62	28.5	7.67	24.3	28.6
AE-OT-GAN	25.2	7.62	17.1	8.24	11.2	7.6

Table 2. The FID and IS between the AE-OT-GAN and the state of the arts on Cifar10 and CelebA.

show the evolution results of the proposed method in Fig. 5 during the GAN's training process. Quality of the generated images increases monotonously during the process. Images in the first four frames of the first row illustrates the results reconstructed from the real latent codes by the generator, with the last frame showing the corresponding ground-truth input images. By examining the frames carefully, it is obvious that as

CT-GAN [49] WGAN-GP [15] WGAN-div [46] WGAN-QC [29] Proposed method

Fig. 6. The visual comparison between the proposed method and the state-of-the-arts on CelebA dataset [50] with ResNet architecture.



WGAN-GP [15] SNGAN [33] WGAN-div [46] AE-OT [2] Proposed method

Fig. 7. The visual comparison between the proposed method and the state-of-the-arts on Cifar10 dataset [22] with ResNet architecture.

the increase of the epochs, the reconstructed images become sharper and sharper, and eventually they are very close to the ground truth. The second row shows the generated images from some generated latent codes (therefore, no corresponding real images). Similarly. the images become sharper as the increase of epochs. Here we need to state that the 0 epoch stage means the images are generated by the original decoder, which are equivalent to the outputs of an AE-OT model [2]. Thus we can conclude that the proposed AE-OT-GAN does improve the performance of AE-OT prominently.

Comparison on CelebA and CIFAR 10 Secondly, we compare with the state-of-the-arts including WGAN-GP [15], PGGAN [18], SNGAN [33], CTGAN [49], WGAN-div [46], WGAN-QC [29] and the recently proposed AE-OT model [2] on Cifar10 [22] and CelebA [50]. Tab. 2 shows the FIDs (lower is better) of our method and the comparisons trained under both the standard and ResNet architectures. The FIDs of other methods come from the listed papers except those of the AE-OT, which are directly computed by our model (the results of epoch 0). From the table we can see that our method gets much better results than others on both the Cifar10 and the CelebA datasets, under both the standard and the ResNet architectures. Also, the generated images of the proposed methods have less flaws compared to other GANs, as shown on Fig. 6 and Fig. 7. The convergence curves of the FIDs for the both datasets can be found in Fig. 8. For the Cifar10 dataset, another popular metric is the Inception score (IS, higher is better), which is also reported on Tab. 2.

Experiment on CelebA-HQ Furthermore, we also test the proposed method on images with high resolution, namely the CelebA-HQ dataset with image size to be 256x256. In our method, the generated images can be treated as locally interpolation among the nearby given real images. In Fig. 9, the left column shows the generated images and the right 5 columns show the top-5 images used to generate them. From Tab. 3, we can see that the performance of the AE-OT-GAN model is better than the

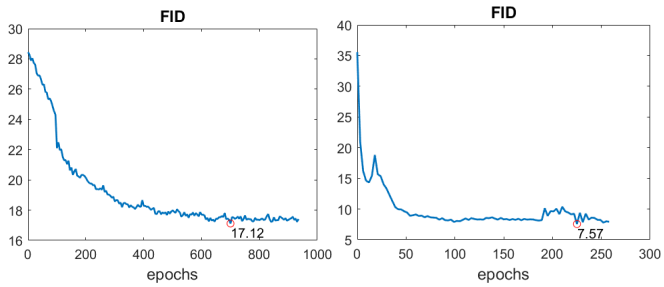


Fig. 8. The FID curves for Cifar10 and CelebA.

comparisons. We also display several generated images in Fig. 10, which are crisp and visually realistic.

PGGAN	WGAN-div	WGAN-QC	AE-OT-GAN
14.7	13.5	7.7	7.4

Table 3. The FIDs of the proposed method and the state-of-the-arts on CelebA-HQ.

Fig. 9. The interpolation of the AE-OT-GAN model. The left column shows the generated images, and the right 5 images are the ones used to generate the left images in the latent space.

5 Conclusion and Future Work

In this paper, we propose the AE-OT-GAN model which composes the AE-OT model and vanilla GAN together. By utilizing the merits of the both models, our method can generate high quality images without mode collapse. Firstly, the images are embedded into the latent space by the autoencoder, then the SDOT map from the uniform distribution to the empirical latent distribution is computed. Sampling from the generated latent distribution by applying the extended SDOT map, we can train our GAN model steady and efficiently. Moreover, the paired latent codes and images give us additional constraints about the generator and help get rid of the mode collapse problem. Using the FID as the metric, we show that the proposed model is able to generate images comparable or better than the state of the arts.

Acknowledgements The project is partially supported by NSF CMMI-1762287, NSF DMS-1737812 and Ford URP and NSFC (61936002, 61772105, 61720106005). -

Fig. 10. The generation results of CelebA-HQ by the proposed method.

References

1. Akash, S., Lazar, V., Chris, R., U., G.M., Charles, S.: Veegan: Reducing mode collapse in gans using implicit variational learning. Neural Information Processing Systems (2017)
2. An, D., Guo, Y., Lei, N., Luo, Z., Yau, S.T., Gu, X.: Ae-ot: A new generative model based on extended semi-discrete optimal transport. In: International Conference on Learning Representations (2020)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML. pp. 214–223 (2017)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019)
5. Burda, Y., Grosse, R., Salakhutdinov, R.: Importance weighted autoencoders. In: ICML (2015)
6. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems (2016)
7. Dai, B., Wipf, D.: Diagnosing and enhancing VAE models. In: International Conference on Learning Representations (2019)
8. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
9. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. In: ICLR (2017)
10. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: International Conference on Learning Representations (2017)
11. Donahue, J., Simonyan, K.: Large scale adversarial representation learning. In: https://arxiv.org/abs/1907.02544 (2019)
12. Figalli, A.: Regularity properties of optimal maps between nonconvex domains in the plane. Communications in Partial Differential Equations **35**(3), 465–479 (2010)
13. Goodfellow, I.: Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 (2016)
14. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets (2014)
15. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NIPS. pp. 5769–5779 (2017)
16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a nash equilibrium (2017)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
18. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: ICLR (2018)
19. Khayatkhoei, M., Singh, M.K., Elgammal, A.: Disconnected manifold learning for generative adversarial networks. In: Advances in Neural Information Processing Systems (2018)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
21. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: NeurIPS (2018)
22. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech report (2009)
23. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric (2016)
24. Lecun, Y., Chopra, S., Hadsell, R.: A tutorial on energy-based learning (01 2006)
25. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), `http://yann.lecun.com/exdb/mnist/`

26. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network (2017)
27. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. arXiv preprint arXiv:1907.11922 (2019)
28. Lin, Z., Khetan, A., Fanti, G., Oh, S.: Pacgan: The power of two samples in generative adversarial networks. In: Advances in Neural Information Processing Systems. pp. 1505–1514 (2018)
29. Liu, H., Gu, X., Samaras, D.: Wasserstein gan with quadratic transport cost. In: ICCV (2019)
30. Lucic, M., Kurach, K., Michalski, M., Gelly, S., Bousquet, O.: Are gans created equal? a large-scale study. In: Advances in neural information processing systems. pp. 698–707 (2018)
31. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research (2008)
32. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
33. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: ICLR (2018)
34. Nijkamp, E., Hill, M., Zhu, S.C., Wu, Y.N.: On learning non-convergent non-persistent short-run mcmc toward energy-based model. arXiv preprint arXiv:1904.09770 (2019)
35. van den Oord, A., Kalchbrenner, N., Espeholt, L., kavukcuoglu, k., Vinyals, O., Graves, A.: Conditional image generation with pixelcnn decoders. In: Advances in Neural Information Processing Systems (2016)
36. van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. In: NeurIPS (2017)
37. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR (2016)
38. Razavi, A., Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2 (2019)
39. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014)
40. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans (2016)
41. Salimans, T., Zhang, H., Radford, A., Metaxas, D.: Improving GANs using optimal transport. In: International Conference on Learning Representations (2018)
42. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
43. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2391–232 (2000)
44. Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein auto-encoders. In: ICLR (2018)
45. Villani, C.: Optimal transport: old and new, vol. 338. Springer Science & Business Media (2008)
46. Wu, J., Huang, Z., Thoma, J., Acharya, D., Gool, L.V.: Wasserstein divergence for gans. In: ECCV (2018)
47. Xiao, C., Zhong, P., Zheng, C.: Bourgan: Generative networks with metric embeddings. In: NeurIPS (2018)
48. Xie, J., Lu, Y., Zhu, S., Wu, Y.: Cooperative training of descriptor and generator networks. IEEE transactions on pattern analysis and machine intelligence (2016)
49. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. In: Advances in Neural Information Processing Systems (2019)

50. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: From facial expression recognition to interpersonal relation prediction. International Journal of Computer Vision (2018)
51. Zhu, S., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. International Journal of Computer Vision (1998)