ENTROPY-REGULARIZED OPTIMAL TRANSPORT GENERATIVE MODELS

Dong Liu, Minh Thành Vu, Saikat Chatterjee, and Lars K. Rasmussen

KTH Royal Institute of Technology, Stockholm, Sweden
E-mail: {doli, mtvu, sach, lkra}@kth.se

ABSTRACT

We investigate the use of entropy-regularized optimal transport (EOT) cost in developing generative models to learn implicit distributions. Two generative models are proposed. One uses EOT cost directly in an one-shot optimization problem and the other uses EOT cost iteratively in an adversarial game. The proposed generative models show improved performance over contemporary models on scores of sample based test.

Keywords: Optimal transport, generative models.

1. INTRODUCTION

Data-driven learning of a probability distribution by a generative model is an important problem in statistical signal processing and machine learning. Recently neural network based generative models are popular tools to study underlying probability distribution of datasets. A prominent example is generative adversarial network (GAN) [1], which learns implicit distribution models.

In the GAN of [1], a generator produces synthetic samples and a discriminator endeavors to distinguish between real samples and synthetic samples. Generators and discriminators are realized using (deep) neural networks. Discriminator and generator play an adversary game against each other using a 'min-max' optimization to learn parameters of neural networks. For generator, the game turns out be minimizing Jensen-Shannon divergence (JSD) between target distribution and induced distribution by generator when discriminator is optimal. Using the same adversary optimization, deep convolutional neural network based GAN (DCGAN) [2] producing good quality synthetic images, has attracted high attention.

JSD has limitations in GANs where generators and discriminators are based on deep neural networks. The first limitation is that back propagation suffers from vanishing gradient. Gradient of cost function with respect to (w.r.t.) generator vanishes as discriminator approaches optimal (see Theorem 2.4 [3]), which stops generator from further learning. The second limitation is due to high sensitivity of JSD to slight perturbations. JSD can be large between a distribution P_x and a distribution $P_{x+\varepsilon}$ where ε is perturbation [3].

Both limitations are addressed in Wasserstein GAN (WGAN) [4]. Wasserstein distance stems from optimal transport (OT) problem, which measures divergence between two distributions. The WGAN formulation does not require an explicit discriminator and it does not has the vanishing-gradient problem. Further, Wasserstein distance/OT is upper bounded by the standard deviation of perturbation ε [3], addressing the second limitation.

OT based cost in WGAN brings a strict constraint to follow in its optimization. Kantorovich-Rubinstein duality used in WGAN requires a supremum over infinite set of all Lipschitz functions with Lipschitz constant equal to one. Various sub-optimal techniques are proposed to enforce the Lipschitz property. An example is weight clipping [4] where neural network parameters (weights) are updated first without Lipschitz constraint and then projected to satisfy Lipschitz constraint in each iteration. Other approaches are gradient penalty [5] and spectrum normalization [6].

In this article, our main contribution is to explore use of Entropy-regularized OT (EOT) cost for generative models. The EOT was studied earlier for efficient comparison between two probability distributions [7]. The major advantage of EOT is that corresponding dual problem is free from Lipschitz constraint. Use of EOT improves analytical tractability allows us to develop two generative models. Our first model considers EOT cost directly on distribution of signals (in our case, on the image pixels). This model uses an one-shot optimization problem, *i.e.* no use of adversarial game in iterations. The second model considers EOT on feature distribution instead of direct signal distribution. In this case we also need to learn a representation mapping, which is implemented as a neural network. This requires alternative optimization of representation mapping and generator. In addition to the above advantage, duality of EOT can be effectively solved and straightforwardly extended to parallel computation.

2. EOT BASED GENERATIVE MODELS

In this section, we begin with Entropy-regularized OT (EOT) cost and then propose generative models.

2.1. Entropy-regularized OT

We denote our working space by $(\mathcal{X}, \|\cdot\|_2)$ where $\mathcal{X} \subset \mathbb{R}^d$ and $\|\cdot\|_2$ is the Euclidean distance. Assume that $\mathcal{X}_1$, $\mathcal{X}_2$ are N-sample subsets of $\mathcal{X}$. Let P be a distribution on $\mathcal{X}_1$ and

Q be a distribution on $\mathcal{X}_2$. OT calculates the minimum cost of transporting distribution P to Q. We use $W(P,Q)$ to denote entropy-regularized OT (EOT) cost as follows:

$$W(P,Q) = \min_{\pi \in \Pi(P,Q)} \langle \pi,\, M \rangle - \lambda H(\pi), \tag{1}$$

where $\langle \cdot,\, \cdot \rangle$ stands for the inner product of two matrices, and $\Pi(P,Q)$ is a set of joint distribution π on the sample sets $\mathcal{X}_1 \times \mathcal{X}_2$ such that π has marginal distributions P and Q. The cost matrix M has elements $[M]_{i,j} = d(x^{(i)}, y^{(j)}) = \|x^{(i)} - y^{(j)}\|_2^2$ and $x^{(i)}, y^{(j)}$ are samples of P, Q, respectively. Here $H(\pi) = \sum_{i,j} -\pi_{i,j} \log(\pi_{i,j})$ and $\lambda \in \mathbb{R}_+$ is the regularization parameter. The entropy regularization in (1) translates to a requirement that the joint distribution π has a high entropy. Note that $\|\cdot\|_2$ is invariant of unitary transform and hence representation of $\mathcal{X}$ in another unitary basis does not change the cost matrix. The duality of EOT cost in (1) is

$$W(P,Q) = \max_{\alpha,\beta \in \mathbb{R}^N} \alpha^T P + \beta^T Q - \sum_{i,j} \lambda e^{\frac{(\alpha+\beta-[M]_{i,j})}{\lambda}}, \tag{2}$$

where α, β are dual variables and $(\cdot)^T$ means transpose. The optimal dual vector β^* of (2) is a subgradient of $W(P,Q)$ with respect to Q. There is a computationally efficient algorithm called Sinkhorn algorithm [7, 8] to solve (2), which alternatively scales the rows and columns of matrix $e^{-\frac{M}{\lambda}}$. This alternative computation gives a pair of vectors $(u, v) \in \mathbb{R}_+^N \times \mathbb{R}_+^N$ that defines the optimal primary and dual variavles (see proposition 2 in [8]):

$$\pi^* = \mathrm{diag}(u)\, e^{\frac{-M}{\lambda}}\, \mathrm{diag}(v), \beta^* = \frac{\log(u^T)\mathbb{1}_N}{N\lambda}\mathbb{1}_N - \frac{\log(u)}{\lambda}. \tag{3}$$

where $\mathrm{diag}(u)$ is a matrix with diagonal entries from vector u and $\mathbb{1}_N$ is a column vector with ones.

2.2. EOT based Generative Models

In this subsection, we propose two generative models. We first develop an EOT based generative model handling signals/data directly. This models is referred to as EOT generative model (EOTGM). In our second model, we use a representation mapping where EOT cost is used to optimize the generative model and representation mapping jointly. The second model is referred as EOT based GAN (EOTGAN).

2.2.1. EOT based generative model (EOTGM)

Assume that P is the unknown true probability distribution of a dataset and Q is a probability distribution induced by a generator $g : \mathcal{Z} \to \mathcal{X}$. Generator g usually is realized by a neural network and maps latent signal $Z \in \mathcal{Z}$ to signal in $\mathcal{X}$, i.e., $g(Z) \in \mathcal{X}$. Latent signal is assumed to follow a fixed distribution $Z \sim P_Z$ (P_Z is usually assumed to be Gaussian). The mapped signal $g(Z) \sim Q$ since g induces Q. Denote the parameter of g by θ. Applying EOT cost to learn Q is equivalent to minimizing $W(P,Q)$ w.r.t. generator g:

$$\operatorname*{argmin}_{g:\mathcal{Z}\to\mathcal{X}} W(P,Q) = \operatorname*{argmin}_{\theta} W(P,Q). \tag{4}$$

Since β^* in (3) is subgradient of $W(P,Q)$ w.r.t. Q, we are able to optimize the generator g such that the induced distribution Q approximates P, using gradient chain rule:

$$\nabla_\theta W(Q,P) = (\nabla_\theta Q)^T \beta^*. \tag{5}$$

Alternatively the optimization problem (4) can be addressed by solving $\mathrm{argmin}_g \langle \pi^*,\, M \rangle$ iteratively using auto-gradient functions in PyTorch [9] or TensorFlow [10], where π^* is primary optimal variable to (1) given by (3). We propose Algorithm 1 to learn distribution P via minimizing the EOT loss w.r.t parameter θ of generator function g.

Algorithm 1 EOT based generative model (EOTGM)

Require: l: the update rate at each iteration, N: the batch size, and θ_0: the initial parameter for g.

1: **while** θ has not converged **do**
2: Sample $\{x^{(i)}\}_{i=1}^N \sim P$, a batch from a real dataset.
3: Sample $\{z^{(i)}\}_{i=1}^N \sim P_z$, a batch of noise samples.
4: Get $\{y^{(i)}\}_{i=1}^N$ by passing $\{z^{(i)}\}_{i=1}^N$ through g.
5: Calculate the cost matrix M.
6: π^*, $\beta^* \leftarrow$ primary and dual solutions of $W(\{x^{(i)}\}_{i=1}^N, \{y^{(i)}\}_{i=1}^N)$ according Equation (3).
7: $\theta \leftarrow \theta - l\,(\nabla_\theta Q)^T \beta^*$. (Or back propagate using loss $\langle \pi^*,\, M \rangle$)
8: **end while**

2.2.2. EOT based GAN (EOTGAN)

In this subsection, we consider representation learning (feature learning) with which usage of EOT is more meaningful than that directly in signal space. It is well-known that Euclidean distance is not well suited to compare two multimedia signals. For example, Euclidean distance between an image and its rotated version can be large, but they are visually same. In Algorithm 1 we construct cost matrix M in EOT using Euclidean distance between real signals and generated signals. Our new proposal is to transform signal through a representation mapping $f : \mathcal{X} \to \mathcal{M}$, $\mathcal{M} \subset \mathbb{R}^m$ and we compare features in the representation space via EOT. We assume that Euclidean distance between features in the representation space is more semantically meaningful. An element of the cost matrix M_f in representation domain (feature domain) is:

$$d_f(x,y) = \|f(x) - f(y)\|_2^2. \tag{6}$$

Our new objective is joint learning of generator g and representation f. A natural question is how to construct f function? Inspired by the triplet loss in [11] aiming at larger distance between distinct classes than in-class distance, we may consider two virtual classes labeled by P and Q. This means that the representation function f should have the algebraic

property: $d_f(x,\tilde{x}) + \gamma \leqslant d_f(x,y)$ for $\gamma > 0$, where x and $\tilde{x}$ are two samples from distribution P and y is a generated signal from distribution Q. Meanwhile, g tries to mitigate this distinction.

Following the above idea, let us denote the distribution of $f(x)$ and $f(y)$ by P_f and Q_f, respectively. Let M_f be the cost matrix in representation domain and its elements $[M_f]_{i,j} = d_f(x^{(i)}, y^{(j)})$, $x^{(i)} \sim P, y^{(j)} \sim Q$. Then we learn f and g using alternative optimization, as follows.

1. Learning of representation f is minimizing EOT cost

$$W(P_f, P_f) = \min_{\widetilde{\pi} \in \Pi(P_f, P_f)} \langle \widetilde{\pi},\, \widetilde{M_f} \rangle - \lambda H(\widetilde{\pi}), \quad (7)$$

where $[\widetilde{M_f}]_{i,j} = d_f(x^{(i)}, \tilde{x}^{(j)})$, $x^{(i)}, \tilde{x}^{(j)} \sim P$, and minimizing EOT cost

$$W(P_f, Q_f) = \min_{\pi \in \Pi(P_f, Q_f)} \langle \pi,\, M_f \rangle - \lambda H(\pi). \quad (8)$$

2. Learning of generator g is minimizing EOT cost

$$W(P_f, Q_f). \quad (9)$$

Both $W(P_f, P_f)$ and $W(P_f, Q_f)$ have lower bounds, but no upper bounds. We combine the step 1 in above using a hinge loss and define the following costs.

$$\begin{aligned} \mathcal{L}_f(P_f, Q_f) &\triangleq \max(0, W(P_f, P_f) - W(P_f, Q_f) + \gamma), \\ \mathcal{L}_g(P_f, Q_f) &\triangleq W(P_f, Q_f), \end{aligned} \quad (10)$$

where $\gamma > 0$. Hinge loss helps to balance the adversarial training of the f and g. Note the our hinge adversarial loss shares similarity only in form to the self-attention GAN [12] and geometric GAN [13] but is motivated differently and defined in different metric. We used neural networks for constructing f and g functions. Let us assume that the parameters of f and g are ω and θ, respectively. Then the adversarial training between representation f and generator g is the following alternative optimization problem:

$$\begin{aligned} &\min_f \mathcal{L}_f(P_f, Q_f) = \min_\omega \mathcal{L}_f(P_f, Q_f), \\ &\min_g \mathcal{L}_g(P_f, Q_f) = \min_\theta \mathcal{L}_g(P_f, Q_f). \end{aligned} \quad (11)$$

The EOTGAN is shown in Algorithm 2.

2.2.3. *Advantage of EOT against OT*

Usage of entropy regularization in EOT avoids the need for Kantorovich-Rubinstein duality of OT, thus is free from Lipschitz constraint. In literature, several methods endeavor to satisfy Lipschitz constraint, for example, projecting neural network parameters into a space fulfilling Lipschitz constraint via weight clipping [4], spectrum normalization [6]), or adding gradient penalty into GAN's cost function [5]. Projecting approaches bring the problem of neural network capacity underuse and limit its ability to learn complex mapping. Gradient penalty approach takes gradients of each layer's weight parameters of a neural network into GAN's cost, thus computation complexity grows fast as the neural network goes deeper. EOT avoids the above mentioned problems and also has the benefit of a lower computation complexity. With entropy-regularization and Sinkhorn algorithm, the computation complexity scales as $\mathcal{O}(N^2)$ [7]. On the other hand, solving OT cost using interior-point methods has computational requirement as $\mathcal{O}(N^3 \log N)$.

Algorithm 2 EOT based GAN (EOTGAN)

Require: l: the update rate at each iteration, N: the batch size and θ_0, ω_0: the initial parameters for g and f.

while θ has not converged **do**
  Sample two batches of data $\{x^{(i)}\}_{i=1}^N, \{\tilde{x}^{(i)}\}_{i=1}^N$, and latent samples $\{z^{(i)}\}_{i=1}^N$, $x^{(i)}, \tilde{x}^{(i)} \sim P, z \sim P_z$.
  Get $\{y^{(i)}\}_{i=1}^N$ by passing $\{z^{(i)}\}_{i=1}^N$ through g.
  $\widetilde{\pi}^* \leftarrow$ solving $W_f\left(\{f(x^{(i)})\}_{i=1}^N, \{f(\tilde{x}^{(i)})\}_{i=1}^N\right)$
  $\pi^* \leftarrow$ sovling $W_f\left(\{f(x^{(i)})\}_{i=1}^N, \{f(y^{(i)})\}_{i=1}^N\right)$
  $\partial f \leftarrow \nabla_\omega \max\left(0, \langle \widetilde{\pi}^*,\, \widetilde{M} \rangle - \langle \pi^*,\, M \rangle + \gamma\right)$
  $\omega \leftarrow \omega - l \cdot \partial f$
  Sample $\{z^{(i)}\}_{i=1}^N$ and get $\{y^{(i)}\}_{i=1}^N$ via g.
  $\pi^* \leftarrow$ sovling $W_f\left(\{f(x^{(i)})\}_{i=1}^N, \{f(y^{(i)})\}_{i=1}^N\right)$
  $\partial g \leftarrow \nabla_\theta \langle \pi^*,\, M \rangle$
  $\theta \leftarrow \theta - l \cdot \partial g$
end while

3. EXPERIMENTAL RESULTS

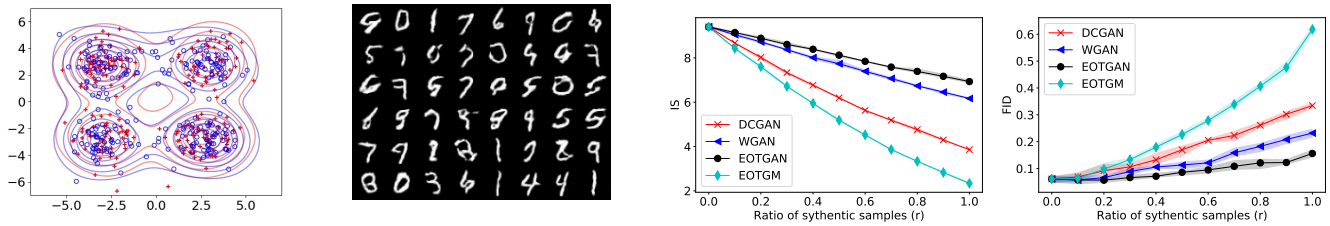
We perform experiments to justify our arguments on loss choice and algorithms. We evaluate our generative models on a toy synthetic dataset of Gaussian-mixture distribution and real image digit dataset MNIST.

3.1. Evaluation Metrics

Inception Score (IS) has been popularly used in evaluation of GAN models [14]. IS is defined as $IS(Q) = \exp\left[\mathbb{E}_{y \sim Q} KL\big(\mathbb{P}(c|y) \| \mathbb{P}(c)\big)\right]$, where $x \sim Q$ indicates synthetic sample from distribution Q induced by generator g, $KL(\cdot,\cdot)$ is Kullback-Leibler divergence, $\mathbb{P}(c|y)$ is the conditional class distribution, and $\mathbb{P}(y) = \int_x \mathbb{P}(c|y)\, dQ(x)$ is the marginal class distribution. Large IS score means generated samples contain clear objects. Generative models with high IS can output high diversity of samples. Apart from KL-based metric, an alternative common metric is Frechet Inception Distance (FID) [15]. FID measures the OT distance of two probability distribution by assuming the two distributions are Gaussian. Smaller FID means the generated samples are more similar to empirical samples. Both FID and IS will be used in our experiments. High IS and low FID are better.

3.2. Evauation of EOTGM using toy dataset

We firstly evaluate our proposed EOTGM on a toy dataset sampled from a known probability distribution: two-



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

Fig. 1. (a) Toy distribution learning (4-mixture Gaussians) using EOTGM. Real samples (red '+') and contour (red curve), versus generated samples (blue 'o') and contour (blue curve) by g. (b) Generated samples by EOTGAN for MNIST dataset. (c) and (d) Comparison of IS and FID (on MNIST) versus mixing ratio r. (For each model at a certain mixture ratio, 5 experiments are independently performed. Each solid curve with markers plots the mean of 5 experiments with shaded areas denoting the range of corresponding results.

dimensional four-mixture Gaussian. This mixture Gaussian is our target distribution to learn, i.e., P. The generator g uses a neural network with structure: Input $\rightarrow$ Dense 256 $\rightarrow$ ReLU $\rightarrow$ Dense 256 $\rightarrow$ ReLU $\rightarrow$ Dense 256 $\rightarrow$ ReLU$\rightarrow$Dense 2. The parameter θ of g here is the set of parameters of this neural network. Latent distribution P_Z used here is standard Gaussian: $\mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1&0\\0&1\end{pmatrix}\right)$. The toy dataset is used by Algorithm 1 (EOTGM) to train g. In Fig.1a, we plot the empirical samples from our toy dataset and the synthetic samples generated by g. The corresponding contours are also plotted. It shows that the induced distribution by g approaches the mixture Gaussian distribution well without missing any mode.

3.3. Evaluation of generative models using MNIST

In this subsection we evaluate both the generative models using MNIST dataset. The representation mapping f in EOTGAN adapts two converlutional layers appended with fully connected layers[1] similar to [16] [17]. Generator g uses the same setting as that of DCGAN and WGAN. Noise P_z is 100-dimensional Guassian. We report IS and FID scores of EOTGAN in comparison with DCGAN and WGAN. Since EOTGAN is trained with representation mapping f that acts as feature mapping, it is not fair to use this representation mapping f to do the evaluation and make comparison since it would gives EOTGAN advantages. Similar to [18], we train a 34-layer ResNet on MNIST to perform feature extraction for metric measurements of IS and FID. In addition, we put EOTGM (Algorithm 1) in the comparison as well.

Data for evaluations is constructed by mixing empirical samples and synthetic samples generated by g. We draw the set $\mathcal{S}_{\text{em}}$ of 2000 empirical samples from MNIST dataset. To generate a set $\mathcal{S}_{\text{syn}}$ of synthetic samples we draw $2000r$ samples from the generator network g where $r \in [0, 1]$ while rest $2000(1-r)$ are sampled directly from MNIST. All following experiments are applied on $\mathcal{S}_{\text{em}}$ and $\mathcal{S}_{\text{syn}}$. The way of mixing empirical data and generated data helps us to identify if a metric is intuitively helpful. Among the chosen metrics IS at $r = 0$ serves as a upper bound for the test while the FID at $r = 0$ serves as a lower bound for the corresponding tests.

IS measures how certain a classifier assigns a class label to a given generated sample. The larger IS is, the better the generative model is. We plot IS versus r for different models in Fig. 1c. IS scores of all four tested models drop with increasing portion of synthetic samples in $\mathcal{S}_{\text{syn}}$, which is consistent with intuition. IS of EOTGAN drops at the slowest rate among the four model as more synthetic samples, for larger r, are mixed into test data. It shows that EOTGAN outperforms WGAN and DCGAN in this test. EOTGM is found to provide the lowest IS. This may be attributed to the setup that EOT optimization with cost measured by Euclidean distance of signals fails to capture semantic similarity.

In Fig. 1d the perfomances of different models are compared using the FID metric. The smaller the FID of a generative model is, the more similar the generated samples are to the empirical samples. EOTGAN is the least affected model among all the four, as the ratio r increases, i.e. the generated samples by EOTGAN is more similar to the empirical ones in the feature space regarding to FID. FID of WGAN is larger than that of EOTGAN. As more generated samples are mixed the FIDs of DCGAN and EOTGM grow even faster, which means the samples generated by these two models are less similar to the empirical samples.

4. CONCLUSION

This work shows that entropy-regularized optimal transport cost is useful to train neural network based generative models for learning implicit probability distributions. With computationally efficient Sinkhorn algorithm, learning of a probability distribution by a generative model can be posed as an one-shot optimization problem. For further progress in quality of generating samples, our experiments show that additional use of representation mapping and alternative optimization based on adversarial game produce better semantic samples.

[1] 32 Conv2d 5×5 $\rightarrow$ PReLU $\rightarrow$ MaxPool 2×2 $\rightarrow$ 64 Conv2d 5×5 $\rightarrow$ PReLU $\rightarrow$ MaxPool 2×2 $\rightarrow$ Dense 256 $\rightarrow$ PReLU $\rightarrow$ Dense 256 $\rightarrow$ PReLU $\rightarrow$ Dense 2

5. REFERENCES

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *ArXiv e-prints*, Nov. 2015.

[3] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks," *ArXiv e-prints*, Jan. 2017.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, et al., "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. Curran Associates, Inc., 2017.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[7] Marco Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 2292–2300. Curran Associates, Inc., 2013.

[8] Marco Cuturi and Arnaud Doucet, "Fast computation of wasserstein barycenters," in *Proceedings of the 31st International Conference on Machine Learning*, Bejing, China, 22–24 Jun 2014, vol. 32 of *Proceedings of Machine Learning Research*, pp. 685–693, PMLR.

[9] "Pytorch autograd," https://pytorch.org/docs/stable/autograd.html.

[10] "Tensorflow automatic differentiation," https://pytorch.org/docs/stable/autograd.html.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823.

[12] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," *ArXiv e-prints*, May 2018.

[13] J. H. Lim and J. C. Ye, "Geometric GAN," *ArXiv e-prints*, May 2017.

[14] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, et al., "Improved techniques for training gans," in *Advances in Neural Information Processing Systems 29*, pp. 2234–2242. Curran Associates, Inc., 2016.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, et al., "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems 30*, pp. 6626–6637. Curran Associates, Inc., 2017.

[16] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, vol. 1, pp. 539–546 vol. 1.

[17] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, June 2006, vol. 2, pp. 1735–1742.

[18] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," *ArXiv e-prints*, June 2018.