Three-Player Wasserstein GAN via Amortised Duality

Nhan Dam[1] , **Quan Hoang**[1] , **Trung Le**[1] , **Tu Dinh Nguyen**[1] , **Hung Bui**[2] and **Dinh Phung**[1]
[1] Monash University
[2] Google DeepMind
{nhan.dam, quan.hoang, trunglm, tu.dinh.nguyen}@monash.edu, buih@google.com, dinh.phung@monash.edu

Abstract

We propose a new formulation for learning generative adversarial networks (GANs) using optimal transport cost (the general form of Wasserstein distance) as the objective criterion to measure the dissimilarity between target distribution and learned distribution. Our formulation is based on the general form of the Kantorovich duality which is applicable to optimal transport with a wide range of cost functions that are not necessarily metric. To make optimising this duality form amenable to gradient-based methods, we employ a function that acts as an amortised optimiser for the innermost optimisation problem. Interestingly, the amortised optimiser can be viewed as a *mover* since it strategically shifts around data points. The resulting formulation is a sequential min-max-min game with 3 players: the generator, the critic, and the mover where the new player, the *mover*, attempts to fool the critic by shifting the data around. Despite involving three players, we demonstrate that our proposed formulation can be trained reasonably effectively via a simple alternative gradient learning strategy. Compared with the existing Lipschitz-constrained formulations of Wasserstein GAN on CIFAR-10, our model yields significantly better diversity scores than weight clipping and comparable performance to gradient penalty method.

1 Introduction

In recent years, deep generative models have become increasingly important in theoretical research and applied machine learning. Crucial to this endeavour is the ability to generate extremely complex and high-dimensional data observed in real-world daily activities such as speech, images, videos and text. An important class of recent deep generative models is the generative adversarial networks (GANs) [Goodfellow *et al.*, 2014] which learn implicit data distribution via a generator $g(z)$ that maps z from an arbitrary space to the data space. Despite its simplicity, GAN has shown an enormous promise in generating high-dimensional data and has been successfully applied to a wide range of applications including 3D object generation [Wu *et al.*, 2016], text-to-image [Zhang *et al.*, 2016], and image-to-image translation [Zhu *et al.*, 2017] to name a few.

Efforts to overcome some fundamental problems in the original GAN formulation, such as mode collapse, have rapidly advanced the theoretical underpinnings of GAN, noticeably f-GAN [Nowozin *et al.*, 2016] , WGAN [Arjovsky *et al.*, 2017], and geometric enclosing networks [Le *et al.*, 2018]. While f-GAN provides an elegant generalisation of GAN to help us better understand the nature of the divergence objective in GAN with the f-divergence, the inherent mode collapse problem still exists [Goodfellow, 2016]. Besides, f-divergence is inherently discontinuous and a slight change in the generator g may lead to a significant change in the f-divergence between the data distribution $\mathbb{P}_d$ and distribution $\mathbb{P}_g$ induced by g [Arjovsky *et al.*, 2017]. This causes difficulty in training generative models involving f-divergence. Another related problem comes from the high dimensionality of the data space, i.e. the true and induced distributions often lie in two separate manifolds, making their supports disjoint and further incurring very large or even infinite f-divergence values [Arjovsky *et al.*, 2017].

One attractive solution to overcome these aforementioned problems in using f-divergence is to employ the *Wasserstein distance* (the most popular special case of *optimal transport cost*), which is inherently continuous and immune against the dimensionality misspecification, to train GAN as proposed in WGAN [Arjovsky *et al.*, 2017]. Given a lower semicontinuous cost function $c : \mathcal{X} \times \mathcal{X} \to [0, +\infty]$, the optimal transport cost between two distributions $\mathbb{P}_d$ and $\mathbb{P}_g$ w.r.t this cost function in its *primal* form is defined as [Villani, 2008; Santambrogio, 2015]:

$$W\left(\mathbb{P}_d, \mathbb{P}_g\right) = \min_{\pi \in \Pi\left(\mathbb{P}_d, \mathbb{P}_g\right)} \mathop{\mathbb{E}}_{(x,y)\sim\pi} \left[c\left(x, y\right)\right], \tag{1}$$

where $\Pi\left(\mathbb{P}_d, \mathbb{P}_g\right)$ represents all couplings π of $\mathbb{P}_d$ and $\mathbb{P}_g$, i.e. a joint measure over $\mathcal{X} \times \mathcal{X}$ with marginal distributions $\mathbb{P}_d$ and $\mathbb{P}_g$. When c is a metric, $W\left(\cdot, \cdot\right)$ in (1) becomes the Wasserstein distance of order 1 between two distributions. Exploiting the convexity of optimal transport (OT), the general Kantorovich *duality* takes the following form [Santambrogio, 2015]:

This work was supported by the Australian Research Council (ARC) DP160109394.

$$W\left(\mathbb{P}_d, \mathbb{P}_g\right) = \max_{f,\tilde{f}} \left\{ \mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f\left(x\right)\right] + \mathop{\mathbb{E}}_{\mathbb{P}_g}\left[\tilde{f}\left(y\right)\right] \right\},$$

where $f, \tilde{f}$ are *bounded* and *continuous* functions satisfying the constraint: $f\left(x\right) + \tilde{f}\left(y\right) \leq c\left(x, y\right), \forall x, y \in \mathcal{X}$. Let $f^c\left(y\right) = \min_x \left\{c\left(x, y\right) - f\left(x\right)\right\}$ be the c-transform of f, one can transform the above optimisation problem into a more computational-friendly form:

$$W\left(\mathbb{P}_d, \mathbb{P}_g\right) = \max_{f} \left\{ \mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f\left(x\right)\right] + \mathop{\mathbb{E}}_{\mathbb{P}_g}\left[f^c\left(y\right)\right] \right\}. \tag{2}$$

Some papers in deep generative models using GAN with Wasserstein distance, e.g. WGAN [Arjovsky *et al.*, 2017] and WGAN-GP [Gulrajani *et al.*, 2017], have attempted to exploit the above duality form, but so far have been strictly limited to the case where the cost function is a norm, i.e. $c\left(x, y\right) = \|x - y\|$. When the cost function is a metric in the underlying space, one can further prove that $f^c = -f$ with $f \in \mathcal{L}_1$ where $\mathcal{L}_1$ denotes the family of 1-Lipschitz functions (with the metric used in the cost function). In summary, WGAN and WGAN-GP attempted to solve:

$$\min_g W\left(\mathbb{P}_d, \mathbb{P}_g\right) = \min_g \max_{f \in \mathcal{L}_1} \left\{ \mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f\left(x\right)\right] - \mathop{\mathbb{E}}_{\mathbb{P}_g}\left[f\left(y\right)\right] \right\}, \tag{3}$$

where g and f are parameterised via neural networks (NNs), but f is required to be in the class of 1-Lipschitz functions $\mathcal{L}_1$. Despite the powerful and elegant formulation of Wasserstein distance for this problem, the restriction on Lipschitz condition unfortunately makes training these models difficult, resulting in weight-clipping heuristic technique in WGAN, regularising through gradient penalty in WGAN-GP and its improved version [Wei *et al.*, 2018] or constraint on the spectral norm (SN) of the weight matrices in SNGAN [Miyato *et al.*, 2018]. In these methods, the family of constrained functions is only a subset of $\mathcal{L}_1$. Model-wise, the strict use of norm as the cost function also arguably limits the true potential of OT.

In this paper, we propose a solution to overcome two aforementioned limitations encountered in the previous work. Our method considers the general use of OT which is valid for a wide range of cost functions that are not necessarily metric, and more importantly we successfully remove 1-Lipschitz constraint. Our key technical contribution is a new formulation that transforms the duality in (2) into a tractable optimisation problem using the amortised optimisation technique. Our new formulation involves a new function approximator which we call *'mover'* $h : \mathcal{X} \to \mathcal{X}$ that attempts to learn the result of the optimisation problem in computing the c-transform. This results in the following new optimisation form which can work with a wide range of cost functions as well as totally removing the Lipschitz constraint:

$$\min_g \max_f \min_h \left\{ \mathop{\mathbb{E}}_{\mathbb{P}_g}\left[f\left(y\right)\right] - \mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f\left(h\left(x\right)\right)\right] + \mathop{\mathbb{E}}_{\mathbb{P}_d}\left[c\left(h\left(x\right), x\right)\right] \right\}. \tag{4}$$

Interestingly, our proposed form can be interpreted as a 3-player game analogy: g is the *generator* as in original GAN; f is the *critic*, taking to be any lower semi-continuous function without 1-Lipschitz constraint; and h is a new player, playing the role of a *mover*. In essence, while the critic tries to fool the generator, the mover and the generator in turn try to fool the critic in different ways. One possible informal setting is to think of the critic f as a police officer who tries to distinguish between the real and counterfeit money, and the generator g as a criminal who makes fake money as in the usual GAN interpretation [Goodfellow *et al.*, 2014], whilst the mover h is a corrupt police officer whose job is to sabotage the officer f to help the criminal g perfect their counterfeit making process. To elaborate, the generator creates a counterfeit from raw materials (i.e. random noise) such that the critic may consider it genuine. Meanwhile, the mover mildly contaminates an authentic object such that the critic is not aware of. At the ideal equilibrium, the critic f becomes the Kantorovich potential (refer to [Villani, 2008; Santambrogio, 2015] for the definition of Kantorovich potential), the mover becomes the identity function $h(x) = x$ which incurs no cost in transport, and the induced distribution $\mathbb{P}_g$ is identical to the data distribution $\mathbb{P}_d$.

In our proposed work, all functions g, f and h are modelled via NNs. An immediate question raised from this formulation is whether the nested min-max-min optimisation in (4) can be addressed in practice. We demonstrate that this can be done via a simple alternative gradient update scheme. This suffices to obtain empirical results that are significantly better than WGAN with weight clipping and comparable to WGAN with gradient penalty on the CIFAR-10 dataset. We also discuss interesting connection between the mover to perturbation training wherein the data are moved randomly. Our results show that having an optimal mover is critical to obtaining good performance.

In summary, our contributions in this paper are threefold: i) we propose a general formulation using OT beyond norm cost for GAN where Lipschitz conditions are all removed, overcoming a key challenge encountered in existing literature of applying Wasserstein distance to train GAN; ii) we introduce a new duality form via the novel use of a *mover* which results in an appealing 3-player game strategy where the new mover attempts to shift the data strategically; and iii) we demonstrate a simple yet effective alternative gradient optimisation strategy for our formulation.

2 Three-Player Wasserstein GAN

We revisit the set up of learning an implicit distribution as follows. The goal is to estimate a distribution over an often high-dimensional space $\mathcal{X}$ where we only have access to its empirical data distribution $\mathbb{P}_d$. If we have prior knowledge that the support of the data distribution lies only in a low-dimensional manifold of $\mathcal{X}$, then we can attempt to directly estimate a mapping g from the coordinate space $\mathcal{Z}$ to $\mathcal{X}$. Formally, let $\mathbb{P}_z$ be a prior over the coordinate space $\mathcal{Z}$, we wish to learn a mapping function $g_\theta : \mathcal{Z} \to \mathcal{X}$, i.e. a deep neural network, in such a way that the pushforward measure $\mathbb{P}_{g_\theta} = g_{\theta \# \mathbb{P}_z}$ is as close to $\mathbb{P}_d$ as possible. In other words, we minimise the OT cost defined in (1) between $\mathbb{P}_{g_\theta}$ and $\mathbb{P}_d$:

$$\theta^* = \arg\min_\theta W\left(\mathbb{P}_d, \mathbb{P}_{g_\theta}\right). \tag{5}$$

Our emphasis is a formulation that can work with any lower semi-continuous cost function c that is not necessarily a metric. Hence, we make use of the general duality in (2) and further parameterise the function f by a neural network f_ϕ.

Combining this with (5) we arrive at the following minimax problem in parametric form:

$$\min_{\theta}\max_{\phi}\left[\mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f_\phi\left(x\right)\right]+\mathop{\mathbb{E}}_{\mathbb{P}_{g_\theta}}\left[f_\phi^c\left(y\right)\right]\right]. \tag{6}$$

2.1 Amortised Optimisation

The generality of the dual form in (6) comes with a serious drawback: the c-transform f_ϕ^c is implicitly defined via the optimisation

$$f_\phi^c\left(y\right)=\min_x\left\{c\left(x,y\right)-f_\phi\left(x\right)\right\} \tag{7}$$

and hence it is not straightforward to optimise or differentiate over the parameters θ and ϕ in (6). To address this difficulty, let us denote $\alpha\left(y\right)=\arg\min_x\left\{c\left(x,y\right)-f_\phi\left(x\right)\right\}$, then $f_\phi^c\left(y\right)=c\left(\alpha\left(y\right),y\right)-f_\phi\left(\alpha\left(y\right)\right)$. We now propose to approximate the minimiser $\alpha\left(y\right)$ via a learnable neural network $h_\psi\left(y\right)$ parameterised by ψ. Formally, the function h is learned by performing the following optimisation

$$\min_\psi \mathop{\mathbb{E}}_{\mathbb{P}_g}\left[c\left(h_\psi\left(y\right),y\right)-f_\phi\left(h_\psi\left(y\right)\right)\right]. \tag{8}$$

Our proposed approach can be viewed from the perspective of *amortised optimisation* as follows. Supposed that we have to solve a large number of optimisation problems. Instead of solving each of them numerically, the idea of amortised optimisation is to estimate a function that maps from the input to an approximate solution of the optimisation problem. Here, instead of solving an optimisation problem for every y to find $\alpha(y)$, we learn the mapping h from the input y to $\alpha(y)$, i.e. the solution of the optimisation problem in (7). We call the amortised optimiser h the *mover* as it attempts to shift elements of the space $\mathcal{X}$.

2.2 Three-Player Game Formulation

Assume that the function class h_ψ where $\psi\in\Psi$ has infinite capacity, then the best mover h_{ψ^*} of the optimisation problem in (8) is clearly $h_{\psi^*}=\alpha$, and so $\mathbb{E}_{\mathbb{P}_{g_\theta}}\left[c\left(h_{\psi^*}\left(y\right),y\right)-f_\phi\left(h_{\psi^*}\left(y\right)\right)\right]=\mathbb{E}_{\mathbb{P}_{g_\theta}}\left[f_\phi^c\left(y\right)\right]$. In practice, h belongs to a finite capacity family of neural networks, thus by definition in (7), $\mathbb{E}_{\mathbb{P}_{g_\theta}}\left[f_\phi^c\left(y\right)\right]\leq\min_\psi\mathbb{E}_{\mathbb{P}_{g_\theta}}\left[c\left(h_\psi\left(y\right),y\right)-f_\phi\left(h_\psi\left(y\right)\right)\right]$. Instead of dealing directly with (6), this leads us to propose the following optimisation problem

$$\min_\theta\max_\phi\left\{\mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f_\phi\left(x\right)\right]+\min_\psi\mathop{\mathbb{E}}_{\mathbb{P}_{g_\theta}}\left[c\left(h_\psi\left(y\right),y\right)-f_\phi\left(h_\psi\left(y\right)\right)\right]\right\}$$

or equivalently,

$$\min_\theta\max_\phi\min_\psi\left\{\mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f_\phi\left(x\right)\right]-\mathop{\mathbb{E}}_{\mathbb{P}_{g_\theta}}\left[f_\phi\left(h_\psi\left(y\right)\right)\right]+\mathop{\mathbb{E}}_{\mathbb{P}_{g_\theta}}\left[c\left(h_\psi\left(y\right),y\right)\right]\right\}. \tag{9}$$

Noting that in (2) we are free to apply the c-transformation to the argument of the first expectation instead. Doing so leads to the following optimisation problem where the mover h is applied only to the empirical data

$$\min_\theta\max_\phi\min_\psi\left\{\mathop{\mathbb{E}}_{\mathbb{P}_{g_\theta}}\left[f_\phi\left(y\right)\right]-\mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f_\phi\left(h_\psi\left(x\right)\right)\right]+\mathop{\mathbb{E}}_{\mathbb{P}_d}\left[c\left(h_\psi\left(x\right),x\right)\right]\right\}. \tag{10}$$

Empirically, we find that (10) has better performance than (9), so this is the form we use in our experiments. We name our proposed model *3-player Wasserstein GAN* (3P-WGAN).

The following discussion is dedicated to (10) but a similar one can be derived straightforwardly for (9). The optimisation problem in (10) is now a min-max-min problem and can be considered as an attempt to solve a 3-player sequential game involving the following 3 players: the generator g, the critic f, and the mover h. As in previous formulations of WGAN and WGAN-GP, the critic f focuses on finding the difference between the learned distribution $\mathbb{P}_g$ and the empirical distribution $\mathbb{P}_d$ by assigning high contrast values to the mismatched regions of the two distributions (i.e. maximising $\mathbb{E}_{\mathbb{P}_g}\left[f\left(y\right)\right]-\mathbb{E}_{\mathbb{P}_d}\left[f\left(h\left(x\right)\right)\right]$). Unlike existing formulations, f in our proposed framework can be a bounded continuous function and *we do not have to deal with the Lipschitz constraint* of f. Another key difference is the presence of the mover h. The mover can attempt to fool the critic by moving the empirical data to reduce the contrast obtained by the critic (i.e. minimising $\mathbb{E}_{\mathbb{P}_g}\left[f\left(y\right)\right]-\mathbb{E}_{\mathbb{P}_d}\left[f\left(h\left(x\right)\right)\right]$) with an optimal moving cost (i.e. minimising $\mathbb{E}_{\mathbb{P}_d}\left[c\left(h\left(x\right),x\right)\right]$). Note that the cost function c acts as a regulariser for the mover, preferring the mover to move as little as possible. It can be observed that the mover h tends to regularise and penalise the extreme critic, which places extreme high values over $\mathbb{P}_g$ and extreme low values over $\mathbb{P}_d$. With the extreme critic, the mover h can perform longer moves to transport $\mathbb{P}_d$ into the high-valued region of the critic, hence reducing the chance for the extreme critic to take over others in the outer maximisation. Certainly, if we ignore the mover by setting it to the identity function (i.e. $h_\psi\left(x\right)=x$), the extreme critic dominates others in the outer maximisation. Consequently, the critic f saturates rapidly, facing the gradient vanishing and making the generator not be further improved.

It can be seen that at the equilibrium point of this game, we obtain $\mathbb{P}_{g_\theta}=\mathbb{P}_d$, the mover is the identity function $h_\psi\left(x\right)=x$, and f_ϕ is the Kantorovich potential.

Despite having 3 nets, our proposed model can still be trained via a simple alternative gradient update scheme, in which the only modification necessary is to update the mover more often than the other two players because of the crucial role of the mover. In particular, for each mini-batch of data, we sequentially update the mover, the critic and the generator, then we iterate this process until convergent or the maximum number of iterations is reached. This learning procedure is summarised in Algorithm 1. Note that we drop the subscripts of g_θ, f_ϕ and h_ψ on lines 4, 6, 8 and 9 in Algorithm 1 to make it succinct since the full notations can be found in (10).

2.3 Remarks

The case of the Wasserstein distance of order p (denoted by W_p) involves using the cost function $c(x,y)=\lambda\left\|x-y\right\|_p^p$ (refer to [Villani, 2008; Santambrogio, 2015]). In this case, (10) becomes (note that we drop the subscripts for parameters to shorten the formula)

$$\min_\theta\max_\phi\min_\psi\left\{\mathop{\mathbb{E}}_{\mathbb{P}_g}\left[f\left(y\right)\right]-\mathop{\mathbb{E}}_{\mathbb{P}_d}\left[f\left(h\left(x\right)\right)\right]+\lambda\mathop{\mathbb{E}}_{\mathbb{P}_d}\left[\left\|h\left(x\right)-x\right\|_p^p\right]\right\}.$$

Since in our formulation the cost acts as the regulariser for the mover, it is interesting to note that the scale of the cost

Algorithm 1 Update scheme of 3P-WGAN.

Input: target distribution $\mathbb{P}_d$, noise distribution $\mathbb{P}_z$, cost function $c(x, y)$, number of mover updates per epoch n_{mover}, batch size m, Adam hyperparameters α, β_1, β_2, network architectures for generator $g_\theta(\cdot)$, critic $f_\phi(\cdot)$ and mover $h_\psi(\cdot)$.
Output: optimal parameter θ for the generator g.

while θ has not converged **do**
  Draw $x_i \overset{\text{iid}}{\sim} \mathbb{P}_d, i = 1, \ldots, m$.
  Draw $z_i \overset{\text{iid}}{\sim} \mathbb{P}_z, i = 1, \ldots, m$.
  $y_i \leftarrow g(z_i), i = 1, \ldots, m$.
  for $t = 1, \ldots, n_{mover}$ **do**
    $\psi \leftarrow \text{Adam}\left(\frac{1}{m}\sum_{i=1}^{m} \nabla_\psi \left[-f(h(x_i)) + c(h(x_i), x_i)\right]\right)$.
  end for
  $\phi \leftarrow \text{Adam}\left(\frac{1}{m}\sum_{i=1}^{m} \nabla_\phi \left[-f(y_i) + f(h(x_i))\right]\right)$.
  $\theta \leftarrow \text{Adam}\left(\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta f(y_i)\right)$.
end while
return θ

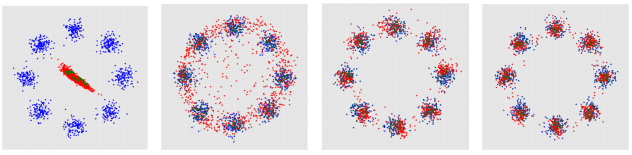
function λ acts like the regularisation strength for h. In the extreme case, as $\lambda \to \infty$, the mover is forced to be an identity function and hence can be removed entirely from our formulation. This corresponds to having no regularisation, and the resulting problem is then the same as the standard WGAN formulation, however f is not constrained to be a 1-Lipschitz function. We expect this would lead to severe overfitting of f, and our empirical results confirm this intuition. As $\lambda \to 0$, the mover is allowed to move more freely, thus increasing the effect of regularisation on the critic. Informally, as the mover has more freedom, the critic f is less likely to be non-smooth since any local non-smoothness can be easily fooled by the mover as it shifts the data around locally.

3 Experimental Results

In this section, we present our experimental results on a synthetic and 2 real-world datasets (i.e. CIFAR-10 [Krizhevsky and Hinton, 2009] and CelebA [Liu *et al.*, 2015]). The synthetic experiment empirically demonstrates the stable convergence property of our proposed 3P-WGAN. On the other hand, experiments on real-world datasets show that 3P-WGAN outperforms WGAN [Arjovsky *et al.*, 2017] and DCGAN [Radford *et al.*, 2015], and yields comparable results to WGAN-GP [Gulrajani *et al.*, 2017]. Then we experiment with different architectures of the mover h to investigate its role. In our proposed 3P-WGAN model, we use the objective function (10) in all experiments. We use TensorFlow [Abadi *et al.*, 2016] and our code is available on GitHub[1].

3.1 Synthetic Experiment

The synthetic samples are generated from a 2-dimensional Gaussian mixture model. There are 8 mixture components with equal mixing proportions. The means of all mixture components are evenly spaced on a circle centred at O with radius of 2. The covariance matrix of each mixture component is $\begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$. The total number of synthetic data points is 2048 and we use full-batch gradient update (since the samples are only 2-dimensional, we can effectively compute full-batch gradient). The architecture of generator, critic and mover is FC - ReLU - FC - ReLU - FC (FC denotes a fully connected layer), with *tanh* applied in the last layer of critic and batch normalisation applied in every layer of generator and critic. The generator is fed with 128 noise units drawn from a uniform distribution. We employ Adam optimiser [Kingma and Ba, 2014] with learning rate of 0.0001 and exponential decay rates β_1, β_2 of $0.0, 0.9$. In each epoch, we update the mover five times while updating generator and critic once. Regarding the reconstruction cost, we use the scaled Euclidean distance $c(x, y) = \lambda \|x - y\|_2$ and anneal the value of λ from 0.1 to 100 over 25,000 epochs.



(a) 100 epochs. (b) 500 epochs. (c) 3000 epochs. (d) 7000 epochs.

Figure 1: Synthetic experiment results: samples from the ground-truth Gaussian mixture model and learned 3P-WGAN. (Blue: real data. Red: generated data. Green: moved real data.)

Figure 1 shows the result of our synthetic experiment. We can see that at the end our proposed model could effectively recover all 8 modes (capturing both mean and covariance) of the Gaussian mixture model. Furthermore, this visualisation shows two points: first our model did converge after roughly 3,000 epochs, and second it stably maintained the convergence after at least 4,000 epochs more, which is even longer than the number of epochs it took to reach convergence.

3.2 Real-World Datasets

We now present our experiments on 2 real-world datasets: CIFAR-10 and CelebA. In CIFAR-10, we use 50,000 colour images belonging to 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each image is of size 32×32. CelebA dataset contains more than 200,000 colour images of celebrities. We use the aligned cropped 64×64 version downloaded from the original website of the dataset. Images in this version focus on the faces.

3.3 Model Architectures

The architectures of our generator and critic follow those in WGAN-GP. However, we apply *tanh* to the last layer and employ batch normalisation in each layer in the critic. The novel component that our proposed model introduces is the mover h. It consists of 4 128-feature map residual blocks that subsequently downscale (the first 2 blocks) and upscale (the last 2 blocks) the input, which eventually goes through a ReLU, a convolutional layer (with kernel size of 3×3, stride of 1, and zero padding) and a *tanh* layer. We also apply batch normalisation in each layer of h. With this architecture, the mover preserves the dimension of the input data.

The total number of epochs in each experiment is 500 yielding roughly 80,000 generator updates, which is fewer than 100,000 generator updates in WGAN-GP. Regarding the

[1] https://github.com/nhandam/p3_wgan

Model	Inception Score	FID
WGAN [Arjovsky *et al.*, 2017]	4.57 ± 0.05	74.6
DCGAN [Radford *et al.*, 2015]	6.40 ± 0.05	37.7
WGAN-GP [Gulrajani *et al.*, 2017]	$\mathbf{7.86 \pm 0.07}$	29.3
3P-WGAN (ours)	7.38 ± 0.08	**28.8**

Table 1: Inception scores (higher is better) and Frechet inception distance (FID) (lower is better) of various models on CIFAR-10.

reconstruction cost, we use the scaled Euclidean distance $c(x, y) = \lambda \|x - y\|_2$ and anneal the value of λ from 0.1 to 100 over 500 epochs. For hyper-parameter selection, we do not tune the parameters based on inception score [Salimans *et al.*, 2016] or Frechet inception distance [Heusel *et al.*, 2017] to avoid overfitting to any metric. We instead try different values of λ, observe the loss of each player and choose the values of λ such that the critic is neither too discriminative nor too flat. We employ Adam optimiser [Kingma and Ba, 2014] with learning rate of 0.0002 and exponential decay rates β_1, β_2 of $0.0, 0.9$. The learning rate is decayed linearly over 100,000 generator updates. Due to its critical role, the mover h is updated 5 times for each weight update of the critic and generator. Other settings include: (i) weights are randomly initialised from Gaussian distribution $\mathcal{N}(0, 0.02\mathbf{I})$ with zero bias; (ii) mini-batch size for training each of 3 players is 64; (iii) and the generator is fed with 128 noise units drawn from a uniform distribution.

3.4 Inception Results

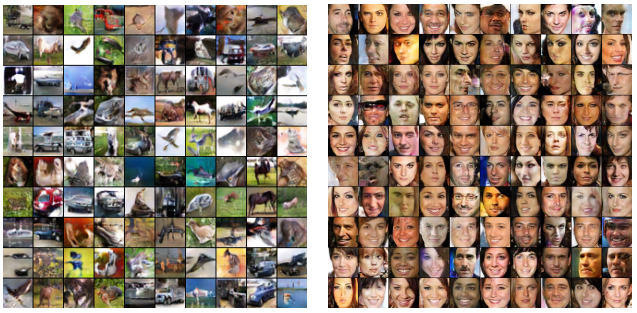
As a quantitative comparison, Table 1 shows the inception scores [Salimans *et al.*, 2016] and Frechet inception distance (FID) [Heusel *et al.*, 2017] of our proposed 3P-WGAN and some prominent GAN models on CIFAR-10 dataset. FIDs are calculated on samples of 50,000 images while inception scores are computed for 10 partitions of 50,000 randomly generated samples. Our proposed 3P-WGAN clearly outperforms WGAN and DCGAN in both criteria. When it comes to WGAN-GP, our 3P-WGAN marginally outperforms its FID but at the same time slightly falls behind its inception score.

3.5 Image Generation

Figure 2 shows samples randomly generated by our proposed 3P-WGAN trained on 2 datasets. Figure 2a shows CIFAR-10 32×32 generated images, in which we can recognise some objects (such as cars, trucks, ships, or horses), whilst generated images of CelebA 64×64 are shown in Figure 2b. Among decent images, we can see variation of some aspects such as gender, hair style, hair colour, facial expression, age, pose angle, moustache and beard, glasses, make-up style. These randomly generated samples from 2 datasets demonstrate that our proposed 3P-WGAN is capable of generating a wide range of decent and recognisable images.

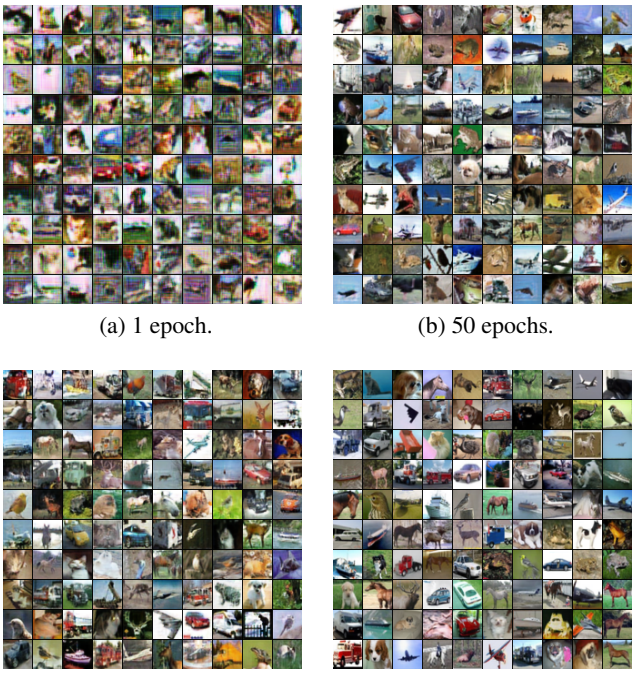
3.6 Inspection of Mover's Behaviours

In this section, we empirically inspect the mover's behaviours. First, in Figure 3 we show the *moved* real images during training of the experiment reported from Section 3.3 to



(a) CIFAR-10 32×32. (b) CelebA 64×64.

Figure 2: Images generated by our proposed 3P-WGAN.



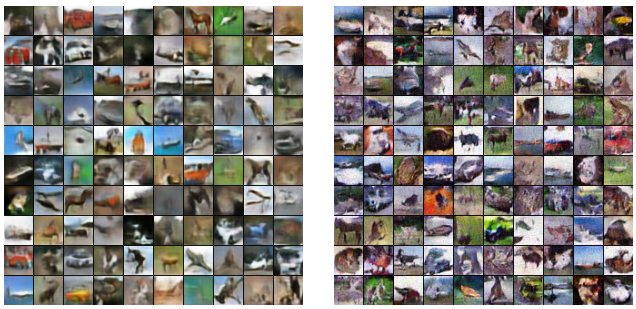
(a) 1 epoch. (b) 50 epochs. (c) 100 epochs. (d) 440 epochs.

Figure 3: Examples of the *moved* real images ($h(x_{real})$) from CIFAR-10 during training.

Section 3.5. To elaborate, each sub-figure is the result when we feed 100 samples randomly selected from CIFAR-10 to the mover after some epochs. We can see that as training proceeds the mover tends to make fewer changes to the images, which supports our claim in Section 2.2 that at the equilibrium the mover is the identity mapping. Second, we vary the architectures of the mover h and conduct experiments that demonstrate not only the importance of the mover but also the relationship between 3 players as discussed in Section 2.2. We use CIFAR-10 in the following experiments.

Mover Is an Identity Function

In this setting, the mover h is no longer a neural net. Instead, we consider 2 scenarios: h is an identity function and h is a noisy identity function (i.e. its output is simply the input added Gaussian noise). The objective function in the first scenario is the same as the objective function of WGAN without



(a) Noise added to the input. (b) Noise added to the last layer.

Figure 4: Generated images of 3P-WGAN trained on CIFAR-10 dataset when the mover h has stochastic noise.

the Lipschitz constraint. The results in both scenarios support the importance of learning the mover h.

In the first configuration, the generated images are very blurry and meaningless, whilst those in the second configuration have only plain patterns and solid colours. In either case, among 100 random samples there are less than 10 distinct scenes, which indicates mode collapse.

Mover Is a Noisy Neural Net

In this setting, the mover h is a neural network. We consider 2 scenarios: Gaussian noise added to the input of h and Gaussian noise added to the last layer of h. Figure 4a and 4b show the generated images of these scenarios respectively. We can see that the samples in Figure 4a are blurry whilst those in Figure 4b are noisy. There is an intuitive explanation for these behaviours. From (10), as training proceeds, the generator g learns to generate images that look like the 'moved real images' (i.e. the result of feeding real images to the mover h), whilst h learns to move real images a short distance and make the critic f consider them as generated images. In the first scenario, h learns to move noisy images to the neighbourhood of real images, thus this process can be loosely considered as denoising, which results in blurry images. As g learns to generate images that look like the output of h, the generated images are blurry as in Figure 4a. In the second scenario where the stochastic noise is added to the output of h, the mover cannot act as a denoiser but only learns to reduce the effect of noise. Consequently, the generated images are noisy as in Figure 4b.

4 Related Work

The work in generative models can be categorised by the divergence/distance used to measure the dissimilarity between target distribution and learned distribution such as Jensen-Shannon (JS) divergence [Goodfellow *et al.*, 2014], f-divergence [Nowozin *et al.*, 2016], maximum mean discrepancy [Dziugaite *et al.*, 2015], Wasserstein distance [Arjovsky *et al.*, 2017], and mixture of various divergences [Hoang *et al.*, 2018; Le *et al.*, 2019; Nguyen *et al.*, 2017].

Among these categories, Wasserstein distance is preferable due to its continuity property and induced weaker topology compared with others [Arjovsky *et al.*, 2017]. Many of existing articles in generative models involving the Wasserstein distance [Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017; Wei *et al.*, 2018] use the Kantorovich duality that requires to enforce the 1-Lipschitz condition over the critic. In particular, the authors of [Arjovsky *et al.*, 2017] proposed to clip the weight matrices of the critic, the authors of [Gulrajani *et al.*, 2017; Wei *et al.*, 2018] used the gradient penalty term to restrict the gradient norm, and the authors of [Miyato *et al.*, 2018] imposed a constraint on the spectral norm of the weight matrices. Weight clipping has been shown to result in low-rank weight matrices for the critic, which therefore uses only a few features [Miyato *et al.*, 2018]. Gradient penalty requires interpolation between the real and generated data, making it hard to evaluate in the high-dimensional setting. The key idea of [Wei *et al.*, 2018] is to impose gradient norm penalty near the data manifold. However, the implementation using Gaussian noise perturbation led to blurry images so they used dropout in the discriminator with debatable assumption that the implicit distance between the input and the perturbed version is constant. Thus we consider this approach as an engineering technique that makes the discriminator robust using dropout instead of imposing Lipschitz constraints. In general, the real families obtained from the techniques in [Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017; Wei *et al.*, 2018; Miyato *et al.*, 2018] are strict subsets of the family of 1-Lipschitz functions, hence only being able to formulate upper bounds of the corresponding Wasserstein distance.

5 Conclusion and Future Work

In this paper, we propose a new formulation for learning GAN using optimal transport based on a general form of the Kantorovich duality. Unlike previous work, our proposed formulation can work with optimal transport built on a wider range of cost functions that are not necessarily metric. The general form of Kantorovich duality appears formidable due to the implicit definition of the function to be optimised as the result of another optimisation problem. Our key contribution in addressing this difficulty is the introduction of an amortised optimiser, learned by another deep neural network. The new network acts as a mover as it strategically shifts the data to fool the critic and sets up a 3-player game between the generator, the critic and the mover. Furthermore, our method does not require imposing any Lipschitz constraints on the critic.

Despite the problem of dealing with a 3-player game, our experimental results demonstrate that with a simple alternative gradient learning strategy, our proposed model can be efficiently trained to achieve comparable results to existing and more restricted WGAN formulations. We experiment with various variants of the mover, including removing it all together, replacing the mover by random noise perturbation, and adding noise to the mover output. The results confirm the importance of training the mover in a strictly adversarial setting without noise.

Our work offers a new perspective to the problem of minimising optimal transport cost in training GANs. Future work can further explore more forms of the cost function (some of which may be specifically designed for a particular task) and the role that the mover plays regarding not only the critic's regulariser, but also the generator's aide (and we need to identify which kind this aide is in particular).

References

[Abadi *et al.*, 2016] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[Dziugaite *et al.*, 2015] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 258–267, Arlington, Virginia, United States, 2015. AUAI Press.

[Goodfellow *et al.*, 2014] 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*, pages 2672–2680, 2014.

[Goodfellow, 2016] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6629–6640, 2017.

[Hoang *et al.*, 2018] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: Training generative adversarial nets with multiple generators. In *International Conference on Learning Representations*, 2018.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[Le *et al.*, 2018] Trung Le, Hung Vu, Tu Dinh Nguyen, and Dinh Phung. Geometric enclosing networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2355–2361. AAAI Press, 2018.

[Le *et al.*, 2019] Trung Le, Quan Hoang, Hung Vu, Tu Dinh Nguyen, Hung Bui, and Dinh Phung. Learning generative adversarial networks from multiple data sources. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

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

[Nguyen *et al.*, 2017] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2670–2680, 2017.

[Nowozin *et al.*, 2016] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[Santambrogio, 2015] Filippo Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, volume 87. Birkhäuser, 2015.

[Villani, 2008] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[Wei *et al.*, 2018] Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of wasserstein gans: A consistency term and its dual effect. *arXiv preprint arXiv:1803.01541*, 2018.

[Wu *et al.*, 2016] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[Zhang *et al.*, 2016] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*, 2016.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.