Learning GANs in Simultaneous Game Using Sinkhorn with Positive Features

RISMAN ADNAN1,3, MUCHLIS ADI SAPUTRA1, JUNAIDILLAH FADLIL1, MARTIANUS FREDERIC EZERMAN2, MUHAMAD IQBAL3, and TJAN BASARUDDIN3

1Samsung Research and Development Indonesia (e-mails: {r.adnan,muchlisin.a,j.fadlil}@samsung.com)
2School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore (e-mail: fredezerman@ntu.edu.sg)
3Department of Computer Science, Universitas Indonesia (emails: {risman.adnan,muhamad.iqbal72,chan}@cs.ui.ac.id)

Corresponding author: Risman Adnan (e-mail: r.adnan@samsung.com).

ABSTRACT Entropy regularized optimal transport (EOT) distance and its symmetric normalization, known as the Sinkhorn divergence, offer smooth and continuous metrized weak-convergence distance metrics. They have excellent geometric properties and are useful to compare probability distributions in some generative adversarial network (GAN) models. Computing them using the original Sinkhorn matrix scaling algorithm is still expensive. The running time is quadratic at $O(n^2)$ in the size $n$ of the training dataset. This work investigates the problem of accelerating the GAN training when Sinkhorn divergence is used as a minimax objective. Let $\mathcal{G}$ be a Gaussian map from the ground space onto the positive orthant $\mathbb{R}^n_+$ with $r \ll n$. To speed up the divergence computation, we propose the use of $c(x,y) = -\varepsilon \log \langle \mathcal{G}(x), \mathcal{G}(y) \rangle$ as the ground cost. This approximation, known as Sinkhorn with positive features, brings down the running time of the Sinkhorn matrix scaling algorithm to $O(rn)$, which is linear in $n$. To solve the minimax optimization in GAN, we put forward a more efficient simultaneous stochastic gradient descent-ascent (SimSGDA) algorithm in place of the standard sequential gradient techniques. Empirical evidence shows that our model, trained using SimSGDA on the DCGAN neural architecture on tiny-coloured Cats and CelebA datasets, converges to stationary points. These are the local Nash equilibrium points. We carried out numerical experiments to confirm that our model is computationally stable. It generates samples of comparable quality to those produced by prior Sinkhorn and Wasserstein GANs. Further simulations, assessed on the similarity index measures (SSIM), show that our model’s empirical convergence rate is comparable to that of WGAN-GP.

INDEX TERMS Entropy regularized optimal transport, generative adversarial network, Sinkhorn divergence, Sinkhorn with positive features.

I. INTRODUCTION

The introduction of the Wasserstein GAN (WGAN) in [1] and its subsequent exposition in [2] propelled optimal transport (OT) to popularity. It has since served as a generic tool to formulate distance metrics and minimax objectives in the Generative Adversarial Network (GAN) framework [3]. The key to using the OT metrics in WGAN formulation lies in the heuristic regularization to enforce a 1-Lipschitz constraint on the discriminator or the critic. The regularization takes several forms, e.g., weight clipping in [1], gradient penalty in [2], and spectral normalization in [4]. Adding a small convex regularization, e.g., an entropic penalty in [5], to the classical linear OT cost is a theoretically sound technique. The entropy regularized optimal transport (EOT) distance and its symmetric normalization, known as the Sinkhorn divergence, are powerful algorithmic tools. They help convexify the minimax objective and come with efficient solvers through the Sinkhorn matrix scaling algorithm. Examples include those proposed in [6]–[9]. They also provide better sample complexity and stability with respect to the input data and the ground cost robustness, as explained, e.g., in [10]–[12].

The GAN framework considers a minimization of a loss functional $\mathcal{L}$ over a parametric family of density $\mu_\theta$. It searches for a solution by computing

$$\min_{\theta \in \Theta} \{ L(\theta) := \mathcal{L}(\mu_\theta, \nu) \}, \quad (1)$$

where $\nu$ is an unknown training data distribution and $\Theta \subset \mathbb{R}^d$ is the feasible domain of the parameter $\theta$. The model distri-
bution is \( \mu_\theta := G_\theta(\xi) \) over a ground set \( X' \subseteq \mathbb{R}^q \). Here, \( \xi \) is a known low-dimensional manifold in the latent space \( Z \subseteq \mathbb{R}^p \) and \( G_\theta : Z \rightarrow X' \) is a push-forward mapping that sends \( \xi \) to \( \mu_\theta \). To solve this minimization problem, GAN employs two neural networks, namely, the generator \( G_\theta \) and the discriminator \( D_\varphi \), with \( \theta \) and \( \varphi \) as their respective weight parameters. An adversarial training of GAN forms a minimax optimization problem in a Sinkhorn distributions whereas \( D_\varphi \) estimates the probability that a sample came from the training data rather than from \( G_\theta \).

In the space of arbitrary functions \( G_\theta \) and \( D_\varphi \), a unique solution exists, with \( G_\theta \) recovering the training data distribution and \( D_\varphi = 0.5 \) everywhere [3].

Let \( S_{\varphi} \) denote the Sinkhorn divergence between two distributions \( \mu_\theta \) and \( \nu \). Let \( c_{\varphi}(x, y) \) be a convex and symmetric transport ground cost in \( \mathcal{X}^2 \subseteq \mathbb{R}^q \). Typical Sinkhorn GANs are formulated with an additional maximization subproblem \( \mathcal{L}(\theta, \varphi) := \max_{\varphi \in \Phi} S_{\varphi}(\mu_\theta, \nu) \). The role of \( D_\varphi \) in a Sinkhorn GAN is no longer as a probability estimator. It becomes a feature map \( D_\varphi : \mathcal{X} \rightarrow Z \) that sends \( \mu_\theta \) to a learned latent space [13]. The transport ground cost, instead of getting fixed to pure Euclidian distances, such as the \( \ell_1 \), \( \ell_2 \), and cosine norms, is encoded by \( D_\varphi \).

Doing so enables \( S_{\varphi}(\mu_\theta, \nu) \) to adversarially distinguish \( \mu_\theta \) and \( \nu \). One can then restate the minimax optimization problem in a Sinkhorn GAN as

\[
\min_{\theta \in \Theta} \max_{\varphi \in \Phi} S_{\varphi}(\mu_\theta, \nu),
\]

where \( G_\theta \) and \( D_\varphi \) represent, respectively, the push-forward and the discriminative feature maps from [13] and [6]. In WGAN literature, with [1], [2], and [4] as prominent examples, these maps are the generator and the critic networks, respectively.

In discrete cases, the Sinkhorn divergence between the model \( \mu_\theta := \frac{1}{n} \sum_{i=1}^n \delta_{\theta_i} \) and the target \( \nu := \frac{1}{n} \sum_{i=1}^n \delta_{\nu_i} \) distributions is the soft distributional minimizer of the EOT distance \( \mathcal{W}_{\varphi} \). Its entropy regularization parameter is \( \varepsilon \rightarrow 0 \).

More detail can be found, e.g., in [5] and [6]. The Sinkhorn matrix scaling algorithm can determine the distance \( \mathcal{W}_{\varphi} \), which measures the total cost of moving all masses from \( \mu_\theta \) to \( \nu \), in a quadratic time \( O(nm) \). The typical choice for the transport ground cost is \( c_\varphi(x, y) := ||D_\varphi(x) - D_\varphi(y)||^p \), the resulting \( \mathcal{W}_{\varphi}(\mu_\theta, \nu)^{1/p} \) is the p-Wasserstein distance when \( p \geq 1 \). When \( p = 1 \), it is the earth-mover distance.

The Sinkhorn GAN can be interpreted as a differentiable minimax game with two players \( G_\theta \) and \( D_\varphi \). The objective \( S_{\varphi}(\mu_\theta, \nu) \) can be either convex or non-convex in \( \theta \), but must be concave in \( \varphi \) (see [6] for an explanation). It is beneficial that the convexity and computational complexity of the Sinkhorn divergence can be controlled by astute choices of the ground cost \( c_\varphi(x, y) \), as shown in [13] and [6]. Good choices include the cosine norm in [7] and [8], the 2-Wasserstein in [6] [14], and the recently-proposed linear-time kernel costs in [12]. We know from [15] and [16] that the stochastic gradient descent-ascent (SGDA) method is commonly deployed to solve the minimax optimization problem in (2). It satisfies the Von Neumann theorem from [17] that reads

\[
\min_{\theta \in \Theta} \max_{\varphi \in \Phi} S_{\varphi}(\mu_\theta, \nu) = \max_{\varphi \in \Phi} \min_{\theta \in \Theta} S_{\varphi}(\mu_\theta, \nu).
\]

For a convex-concave (CC) objective, a vast literature on the simultaneous SGDA (SimSGDA) as a solution to the problem is available. The treatments in [18]–[21] are but a few examples. Under the assumption that the objective is locally strongly-convex in \( \theta \) and strongly-concave in \( \varphi \), SimSGDA has been shown to linearly converge to a local Nash equilibrium in [22]–[24]. At each iteration, it performs a gradient-descent along \( \theta \) and a gradient-ascent along \( \varphi \) with an equal step size or learning rate. Irrespective of which player updates its parameters first, SimSGDA satisfies the Von Neumann minimax theorem in (3).

Despite theoretical progress on SimSGDA, prior Sinkhorn GAN implementations in [6], [7], [9] and [12] carried out the more expensive sequential game. It is solved by a sequential SGDA (SeqSGDA), in which \( D_\varphi \) updates its parameters several times, either sequentially or in an alternate manner, before \( G_\theta \) does. It is known theoretically from [15], [18] and demonstrated empirically in [13] that SeqSGDA is more suitable for a nonconvex-nonconcave (NCNC) objective.

In this work, we accelerate the Sinkhorn GAN training based on two strategies. First, we use SimSGDA, instead of SeqSGDA, to efficiently solve the minimax optimization. Under some mild conditions on the ground cost adversarial from [11], we hypothesize a solution that convergences. Second, we deploy the linear-time Sinkhorn Algorithm with a Gaussian kernel from [12] to compute \( \mathcal{W}_{\varphi} \). Our strategies reduce the running time from \( O(nm) \), which is quadratic in \( n \), to \( O(rn) \), which is linear in \( n \) since \( r \) is a constant. In addition to the gain in training speed, our approach learns with a better generalization than the one proposed in [25]. By extensive simulations we demonstrate empirically that our model, with the CC objective, achieves a local Nash equilibrium efficiently. The main tools are mini-batch approximation, the 1st order SimSGDA algorithm, and a spectral normalization on \( D_\varphi \). Our work aligns well with recent game-theoretical studies in [18] and [15] that interpret the minimax optimization in GAN, with the CC objective, as a simultaneous game solvable by SimSGDA.

**Notation.** The transpose of a matrix \( A \) is \( A^\top \). The canonical inner product (or Frobenius dot-product) of vectors (or matrices) \( u \) and \( v \) denoted by \( \langle u, v \rangle \), is \( \sum u_i \cdot v_i \). The direct sum and the Kronecker product of dual matrices \( a \) and \( b \) are \( a \oplus b \) and \( a \otimes b \), respectively.

Given the space \( \mathbb{R}^d \), we use \( \mathcal{X} \) and \( \mathcal{Y} \) to denote bounded subsets. A set of probability distributions over \( \mathcal{X} \) is denoted by \( \mathcal{M}_+(\mathcal{X}) \) for a given \( \mathcal{X} \subseteq \mathbb{R}^d \).

The notation \( T : \mathcal{M}_+(\mathcal{Z}) \rightarrow \mathcal{M}_+(\mathcal{X}) \) represents a linear map from \( Z \) to \( X \). For any \( x \in \mathcal{X} \), we use \( \delta_x \) to denote the Dirac (unit mass) distribution at \( x \).

The uniform histogram \( \{1/m, \ldots, 1/m\} \) in \( \mathbb{R}^n \) is represented by \( \mathbf{1}_m \).

For an \( n \times m \) real-valued matrix \( P \), that is, \( P \in \mathbb{R}^{n \times m} \), the row average
and the column average are written as $P_{m} \in \mathbb{R}^{m}$ and $P^{T} \in \mathbb{R}^{n}$, respectively.

Let $f : \mathbb{R}^{d} \rightarrow \mathbb{R}$ be a given real-valued function. Then $\nabla f$ is its gradient and $\nabla^{2} f$ is its Hessian. When $f$ takes in two variables $x \in \mathbb{R}^{d_1}$ and $y \in \mathbb{R}^{d_2}$, that is, $f : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \rightarrow \mathbb{R}$, then $\nabla_{x} f$ and $\nabla_{y} f$ are the partial gradients with respect to $x$ and $y$, respectively. Its partial Hessian matrices are $\nabla^{2}_{x} f$, $\nabla^{2}_{x y} f = \nabla^{2}_{x} f$ and $\nabla^{2}_{y} f$. We write $f \in \mathcal{O}(g)$ if $f \leq Cg$ for a universal constant $C$.

A. CONTRIBUTIONS

Our study yields the following contributions:

- A proposal of an efficient and stable SimSGDA algorithm to train Sinkhorn GANs to stationary points. They converge exactly to local Nash equilibriums when using the convex-concave objective.
- An empirical demonstration that simultaneously minimizing the asymmetric distance $W_{\phi}^{\epsilon}(\mu_{\theta}, \nu)$ and maximizing the symmetric divergence $S_{\phi}^{\epsilon}(\mu_{\theta}, \nu)$ still lead to a stable Sinkhorn GAN training.
- A practical numerical implementation to illustrate the versatility of Sinkhorn with positive features. By learning adversarially a Gaussian kernel induced from a positive feature map, the running time of the Sinkhorn Algorithm goes from quadratic down to linear time.
- A structural similarity index measurement on the non-asymptotic convergence rate. We show that our model has a comparable convergence rate to that of WGAN-GP. Both models output samples which are on par with one another in quality.

B. RELATED WORKS

Table 1 lists prominent GAN models alongside ours, for comparative purposes. The models are organized based on their respective minimax objective formulations and optimization techniques.

**Formulation.** The first model that works in large-scale GANs is SGD-AutoDiff [6]. It uses the Sinkhorn divergence as the minimax objective and an MMD-like 2-Wasserstein ground cost. Another model is OT-GAN [7]. It uses a mini-batch energy (MBE) distance and the cosine ground cost. EOT-GAN [9] uses the same ground cost as SGD-AutoDiff, but a different minimax objective by combining the Sinkhorn divergence and the Hinge loss. Yet another model, SWGAN [8], uses the Sinkhorn divergence as the minimax objective, validated on both the 1-Wasserstein and encoded cosine ground costs.

To solve the minimax optimization problem in Sinkhorn GANs, our model uses both the asymmetric $W_{\phi}^{\epsilon}(\mu_{\theta}, \nu)$ distance and symmetric $S_{\phi}^{\epsilon}(\mu_{\theta}, \nu)$ divergence for minimization and maximization objectives, respectively. A recent model, SiNG [26], uses the Sinkhorn divergence as a minimax objective with a natural gradient. SiNG adopts the squared $\ell_{2}$ ground cost defined by $D_{\phi}$. All of these prior models use the so-called vanilla Sinkhorn matrix scaling algorithm, whose running time is quadratic [5]. In contrast, we use the recently-proposed Sinkhorn with positive features (SiPF). It uses the Gaussian kernel as the map to achieve a linear time performance [12].

**Optimization.** A variety of 1st order gradient-based techniques are utilized by prior models. SGD-AutoDiff implements a sequential SGDA (min-max optimization) with a weight clipping regularization. OTGAN utilizes an alternating SGDA (min-max optimization) without any additional regularization. Instead of using SGD, EOT-GAN proposes a simultaneous SGD in a min-min optimization context. Unfortunately, EOT-GAN performs poorly on higher-resolution datasets and shows instability during training. SWGAN performs an oracle-based non-convex SGD with a proven theoretical convergence guarantee to a stationary distribution, provided that the discriminator can be shown to be approximately optimal.

Our model performs SimSGDA that minimizes $W_{\phi}^{\epsilon}(\mu_{\theta}, \nu)$ and maximizes $S_{\phi}^{\epsilon}(\mu_{\theta}, \nu)$ simultaneously. It has an equal gradient step size for both the push-forward map $G_{\theta}$ and the discriminative feature map $D_{\phi}$. Instead of implementing the SiPF model in a sequential or alternating SDGA, which had been done in [12], we opt for the simultaneous one. Implementing either of the earlier approaches requires several expensive critic updates for $D_{\phi}$, which is also commonly found in variants of OT-GAN [7]. Our novel optimization approach, via Algorithm 2 below, is stable and more efficient.

II. PRELIMINARIES

We recall necessary notions and known results on EOT, paying special attention to Sinkhorn divergence’s desirable
properties. These include smoothness, positivity, convexity, and differential properties.

A. EOT PRIMAL FORMULATION

Let the model and target distributions be

\[ \mu_\theta := \frac{1}{m} \sum_{i=1}^{m} \delta_{x_i} \quad \text{and} \quad \nu := \frac{1}{n} \sum_{i=1}^{n} \delta_{y_i}. \]  

(4)

Given a constant entropy regularization parameter \( \varepsilon \to 0 \), the Sinkhorn divergence between \( \mu_\theta \) and \( \nu \) is

\[ S_\varepsilon^\varphi (\mu_\theta, \nu) = \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) - \frac{1}{2} \left( \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \mu_\theta) + \mathcal{W}_\varepsilon^\varphi (\nu, \nu) \right). \]  

(5)

The distance \( \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) \) in \( \mathcal{M}_+^+ (\mathcal{X}) \times \mathcal{M}_+^+ (\mathcal{X}) \to \mathbb{R}_+ \) is

\[ \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) := \min_{P \in \mathbb{R}_{++}^{m \times n}} \langle P, C \rangle - \varepsilon H (P) + \varepsilon. \]  

(6)

The transport operators are \( A := P L_m \) and \( B := P^T L_n \). The transport ground cost is \( C := [c_{\varphi} (x_i, y_j)]_{i,j} \) and \( H \) is the Shannon entropy \( H (P) := - \sum_{i,j} P_{i,j} \log P_{i,j} - 1 \). The EOT adds an entropy scheme to produce a smooth and \( \varepsilon \)-strongly convex distance with a unique solution \( P^* \). It decomposes into

\[ P^* = \text{diag}(u^*) K \text{ diag}(v^*), \]  

(7)

for some optimal scalings \( u^* \in \mathbb{R}_+^m \) and \( v^* \in \mathbb{R}_+^n \), with \( K := \exp (-C/\varepsilon) \) being the associated Gibbs kernel. In this formulation, we can use a fixed point iteration on a modern GPU, known as the Sinkhorn matrix scaling algorithm [5], to determine \( u^* \) and \( v^* \). Once the transport ground cost is fixed, the task entails initializing \( u \) and \( v \) to any arbitrary positive vectors and, then, applying the Sinkhorn Algorithm.

B. EOT DUAL FORMULATION

The dual formulation comes directly from the application of the Lagrange multipliers \( \alpha \) and \( \beta \) and the Legendre transformation. As an unconstrained maximization problem, it is summarized from [27] and [28] in the following proposition. A proof has been supplied in [13, Section II.B].

Proposition II-B.1. The EOT primal formulation in (6) is equivalent to a dual formulation

\[ \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) := \max_{\alpha \in \mathbb{R}_{++}^m, \beta \in \mathbb{R}_{++}^n} \left\{ \langle \mu_\theta, \alpha \rangle + \langle \nu, \beta \rangle - \varepsilon \left( \mu_\theta \otimes \nu \exp \left( (\alpha + \beta - c_\varphi) / \varepsilon \right) - 1 \right) \right\}. \]  

(8)

It sends \( (x, y) \in \mathbb{X}^2 \to (x \beta + y \gamma) \). The dual formulation is also equivalent to a maximization problem

\[ \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) := \max_{\alpha \in \mathbb{R}_{++}^m, \beta \in \mathbb{R}_{++}^n} \mathbb{E}_{\mu_\theta \otimes \nu} [F_\varepsilon (\alpha, \beta)] + \varepsilon, \]  

(9)

where \( F_\varepsilon (\alpha, \beta) := \alpha \beta + \varepsilon \exp \left( (\alpha + \beta - c_\varphi) / \varepsilon \right) \). There exists a primal-dual relationship

\[ \pi = \exp \left( (\alpha + \beta - c_\varphi) / \varepsilon \right) \cdot (\mu_\theta \otimes \nu). \]  

(10)

It links an OT plan to an optimal \( (\alpha, \beta) \) that solves (8).

When \( \varepsilon > 0 \), it is clearly beneficial to use the EOT dual formulation since it can be rewritten as the maximization of an expectation with respect to the product measure \( \mu_\theta \otimes \nu \). The dual problem is unconstrained and jointly concave in both variables. This allows us to fix one and optimize over the other.

Let \( T \) be the Sinkhorn mapping

\[ T: (\mu_\theta, \alpha) \to \left( y \in \mathbb{X} \to \min_{x \sim \mu_\theta} \varepsilon [c_\varphi (x, y) - f (x)] \right), \]  

(11)

with the SoftMin operator

\[ \min_{x \sim \mu_\theta} \varepsilon \gamma (x) := - \varepsilon \log \int_{\mathbb{X}} \exp (-\varepsilon (x) / \varepsilon) d\mu_\theta (x). \]  

(12)

Let \( (\alpha, \beta) \) satisfy \( \alpha = T (\nu, \beta) \) and \( \beta = T (\mu_\theta, \alpha) \). Then the EOT dual formulation is weak continuous and differentiable. Its gradient is \( \nabla \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) = (\alpha, \beta) \).

C. EOT SEMI-DUAL FORMULATION

We focus on \( \alpha \) as a function of \( \beta \) in \( \alpha = T (\nu, \beta) \) and vice versa in \( \beta = T (\mu_\theta, \alpha) \). The dual formulation can then be interpreted as a smoothed version of the \( c \)-transform that links the dual potentials. A commonly used \( c \)-transform of \( \beta \) is \( \beta^c (x) := \min_{y \in \mathbb{Y}} [c_\varphi (x, y) - \beta (y)] \). In EOT, however, we make use of the \( c, \varepsilon \)-transform

\[ \beta^{c, \varepsilon} (x) := - \varepsilon \log \int_{\mathbb{X}} \exp \left( (\beta (x) - c_\varphi (x, y) / \varepsilon \right) d\mu_\theta (x). \]  

(13)

One sees \( \beta^{c, \varepsilon} \) as a smoothed version of \( \beta^c \), which depends on \( \mu_\theta \). We omit \( \mu_\theta \) from the notation for brevity. The transform leads to a semi-dual EOT formulation [30].

Proposition II-C.1. The EOT primal formulation between \( \mu_\theta \) and \( \nu \) in (8) is equivalent to a semi-dual formulation

\[ \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) := \max_{\beta \in \mathbb{C}(\mathcal{X})} \langle \mu_\theta, \beta^{c, \varepsilon} \rangle + \langle \nu, \beta \rangle. \]  

(14)

This semi-dual formulation is equivalent to a maximization of an expectation with respect to only one of the marginals

\[ \mathcal{W}_\varepsilon^\varphi (\mu_\theta, \nu) := \max_{\beta \in \mathbb{C}(\mathcal{X})} \mathbb{E} [\beta^{c, \varepsilon} (\beta)]. \]  

(15)

In discrete cases, \( \beta^{c, \varepsilon} (\beta) \) is given by

\[ - \varepsilon \log \left( \sum_{i=1}^{n} \exp \left( \frac{\beta_i - c_\varphi (x, y_i)}{\varepsilon} \right) \right) + \sum_{i=1}^{n} \beta_i \nu_i. \]  

(16)

Proof. One replaces \( \alpha \) by \( \beta^{c, \varepsilon} \) in the dual formulation in (8) and proceeds accordingly. \( \Box \)

Remark II-C.1. The potential \( \beta \) of a discrete measure \( \nu \) with \( n \)-Diracs is an \( n \)-dimensional vector. If \( \varepsilon > 0 \), then the gradient is \( \nabla_{\beta} B^{c, \varepsilon} (\beta) = \nu - \mathcal{X} \) and the Hessian is

\[ \partial_{\beta}^2 B^{c, \varepsilon} (\beta) = \frac{1}{\varepsilon} (\mathcal{X}_i (x) \mathcal{X}_i (x)^T - \text{diag}(\mathcal{X}_i (x))) \]  

(17)

with

\[ \mathcal{X}_i (x) := \frac{\exp \left( \frac{\beta_i - c_\varphi (x, y_i)}{\varepsilon} \right)}{\sum_{j=1}^{n} \exp \left( \frac{\beta_j - c_\varphi (x, y_j)}{\varepsilon} \right)}. \]  

(18)
Since it can further be shown that $0 \leq \partial^2_{\beta} B^2_\varepsilon (\beta) \leq \frac{1}{2}$, we know that $B^2_\varepsilon$ is convex with a Lipschitz gradient. A strong convexity, however, requires a strictly positive lower bound on the eigenvalues of the Hessian. The semi-dual formulation is convex but not strongly convex. The lower bound on the eigenvalues of the Hessian is 0 [30]. Convexity is an important success criteria in solving the EOT via gradient descent optimization techniques [15], [18], [31].

III. MAIN CONTRIBUTIONS

Instead of the quadratic cost Sinkhorn matrix scaling algorithm, we equip our GAN implementation with a linear-time iteration by using Sinkhorn with positive features. This section explains the key approximations in developing the SimSGDA Algorithm, before completing the model.

A. SINKHORN WITH POSITIVE FEATURES

Computing $\mathcal{W}_c^\varepsilon (\mu, \nu)$ is typically done by the Sinkhorn matrix scaling algorithm. It requires the convex ground cost $c_\varepsilon (D_\varphi (x), D_\varphi (y))$ to embed a sample $(x, y) \in \mathcal{X}^2$ into a latent space [13]. It is common to define the convex ground cost first before the kernel $\mathbf{K} := \exp(-\mathbf{C}/\varepsilon)$.

Computing SimSGDA Algorithm, before completing the model.

Proposition III-A.1. Let $I$ be the identity matrix of the required dimension. Let $d > 1$, $\varepsilon > 0$, and $(x, y)$ be the kernel on $\mathbb{R}^d$ such that, for all $x, y \in \mathbb{R}^d$, $k(x, y) = e^{-\|x-y\|^2/\varepsilon}$. Let $W_0^d$ be the Lambert function. Let $R > 0$, $q = \frac{2dW_0(R^2/\varepsilon)}{\sigma^2}$, and $a^2 = qe/4$. Let the random variable $\rho$ be normally distributed, with $\rho = N(0, \sigma^2 I)$. For all $x, u \in \mathbb{R}^d$, we define the map

$$\mathcal{G}_\phi (x, u) := (2q)^{d/4} e^{-2\varepsilon^{-1}\|x-u\|^2} e^{-\left(\frac{-1+\sqrt{1+4\varepsilon^2}}{2\varepsilon^2}ight)}.$$ (24)

Then, for any $x, y \in \mathbb{R}^d$, we have

$$k(x, y) = \int \mathcal{G}_\phi (x, u) \mathcal{G}_\phi (y, u) d\rho(u).$$

Proof. The proposition can be proven by rewriting the right hand side of (24) before following the route explained in [12, Appendix A.4].

Choosing the Gaussian kernel feature map improves our Sinkhorn GAN in two directions. First, it becomes memory efficient. The computation of the gradient of $\mathcal{G}_\phi$ is independent of the computation required by the Sinkhorn Algorithm. Second, computing the Sinkhorn divergence $S^\varepsilon_\phi (\mu, \nu)$ to a high accuracy can be done even when $\varepsilon \rightarrow 0$. The main difference between our Sinkhorn GAN implementation and the SiPF model in [12] is that we adopt the key approximations from our prior work [13]. The work improved on a version of SGD-AutoDiff but not of the OT-GAN [7]. Furthermore, Scetbon et al. train their model by using SeqSGDA, in contrast to our SimSGDA proposal.

### Algorithm 1 Sinkhorn Algorithm with Positive Features

**Input:** $K, a, b, L$

**Output:** $u, v$

1: for $i = 1, \ldots, L$ do
2: $v \leftarrow b/K^\top u$
3: $u \leftarrow a/Kv$
4: end for

Given a fixed ground cost function $c$ on $\mathbb{R}^d$, we can redefine a parametric ground cost on $\mathcal{X}$ as

$$c_\varphi (x, y) := c \circ h_\varphi (x, y) := c (D_\varphi (x), D_\varphi (y)).$$ (21)

These maps enable us to build the corresponding positive-definite kernel

$$k_\varphi (x, y) := \langle \mathcal{G}_\phi (x), \mathcal{G}_\phi (y) \rangle := \mathcal{G}_\phi (x)^\top \mathcal{G}_\phi (y).$$ (22)

As a by-product and since $D_\varphi$ is positive, we can define, for all $(x, y) \in \mathbb{R}^d$, the ground cost function

$$c_\varphi (x, y) := - \varepsilon \log \mathcal{G}_\phi (D_\varphi (x))^\top \mathcal{G}_\phi (D_\varphi (y)).$$ (23)

Although the mapping kernel $\mathcal{G}_\phi$ can be chosen arbitrarily, this work implements the Gaussian kernel, whose associated ground cost is the $\ell_2$-norm Euclidean metric [12], already used by previous Sinkhorn GANs in [6], [9]. Our $\mathcal{G}_\phi$ feature map implementation is based on [12, Lemma 1] with $\phi$ initialized with a normal distribution.

VOLUME 4, 2016
B. LEARNING GANS WITH SIMSGDA ALGORITHM

Our focus is on the minimax optimization with a CC objective derived from the convex kernel ground cost \( c_{\kappa} \) in (8). A variant of the SimSGDA Algorithm can efficiently find an approximate local Nash equilibrium [15], [16]. To find a local Nash equilibrium in SimSGDA, it is typical to have \( G_\theta \) and \( D_\varphi \) randomly initialize their variables and follow their respective gradients, computed by using the reverse AutoDiff [32]. That is, at each step \( n = 1, 2, \ldots \), each player updates

\[
\begin{align*}
\theta_{n+1} &\leftarrow \theta_n - \eta \nabla_\theta S_{c_{\kappa}}^{\varepsilon}(\theta_n, \nu_n), \\
\varphi_{n+1} &\leftarrow \varphi_n + \eta \nabla_\varphi S_{c_{\kappa}}^{\varepsilon}(\theta_n, \nu_n).
\end{align*}
\]

(25)

\( G_\theta \) descends a gradient on \( S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \) while \( D_\varphi \) ascends the gradient with the same gradient step or learning rate \( \eta \). Algorithm 2 is our SimSGDA implementation. It comes with additional heuristic approximation that, as \( \varepsilon \to 0 \), sets \( \nabla_\theta V_{c_{\kappa}}^{L}(\theta, n, \varphi_n) \approx 0 \) and \( \nabla_\varphi V_{c_{\kappa}}^{L}(\theta_n, \nu) \approx 0 \) to the generator update in each step. This turns (25) into

\[
\begin{align*}
\theta_{n+1} &\leftarrow \theta_n - \eta \nabla_\theta W_{c_{\kappa}}^{\varepsilon}(\theta_n, \nu_n), \\
\varphi_{n+1} &\leftarrow \varphi_n + \eta \nabla_\varphi S_{c_{\kappa}}^{\varepsilon}(\theta_n, \nu_n).
\end{align*}
\]

(26)

The approximation intuitively implies that small changes in \( \theta \) result in negligible changes in the corresponding OT plan between points in the \( \mu_\theta \) and \( \nu \) distributions [13]. The formal expression is in [8, Theorem 3.1]. Game theoretically, \( G_\theta \) can take a different course of action, say, minimizing the asymmetric \( W_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \), instead of \( D_\varphi \). The latter maximizes the symmetric \( S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \) to compute the discriminative feature map. We leave a theoretical investigation into this approximation for a future work.

**Algorithm 2 SimSGDA Algorithm**

**Input:** the real dataset \((y_j)_{j=1}^m\) and the hyperparameters (mini-batch size \( m \), learning rate \( \eta \), number of Sinkhorn iteration \( L \), and entropy regularization weight \( \varepsilon \))

**Output:** \( \theta, \varphi \)

**Initialization:** \( \theta \leftarrow \theta_0, \quad \varphi \leftarrow \varphi_0 \)

1. **while** \( \theta \) has not converged **do**
2. **Sample** \((y_j)_{j=1}^m\) from the real dataset
3. **Sample** \((z_i)_{i=1}^m \sim \zeta\)
4. **Compute** \((x_i)_{i=1}^m := G_\theta(z_i^m)\)
5. **\( \varphi \leftarrow \text{AutoD}_D \left( S_{c_{\kappa}}^{(L)}(x_1^m, y_1^m) \right) \)**
6. **\( \varphi \leftarrow \varphi + \eta \text{ADAM}(\varphi) \)** with Spectral Norm
7. **\( \theta \leftarrow \text{AutoD}_\theta \left( V_{c_{\kappa}}^{(L)}(x_1^m, y_1^m) \right) \)**
8. **\( \theta \leftarrow \theta - \eta \text{ADAM}(\varphi) \)**
9. **end while**

We compute the gradient at each step and approximate \( \nabla_\theta S_{c_{\kappa}}^{\varepsilon}(\mu_\theta, \nu) \) by \( \nabla_\theta S_{c_{\kappa}}^{X_{c_{\kappa}}^{L}}(\mu_\theta, \nu, \hat{v}_m) \) in mini-batch sampling. Thus, we can optimize \( \min_{\theta} S_{c_{\kappa}}^{\varepsilon}(\mu_\theta, \nu) \). Another approach is to use spectral normalization (SN) to enforce the \( K \)-Lipschitz constraint on each discriminator layer [4]. SN remains independent of the objective function and the Sinkhorn algorithm since it is a matrix operation.

The SimSGDA hyperparameters \((m, \eta, \varepsilon, L)\) are the mini-batch size \( m \), the learning rate \( \eta \), the regularization parameter \( \varepsilon \), and the number \( L \) of Sinkhorn iteration. Note that the procedure \( \text{AutoD}_D \) corresponds to the classical reverse-mode automatic differentiation of the \( L \) steps of the Sinkhorn iteration. Its running time is, therefore, \( O(L m n) \). The training procedure consists of the same optimization steps to train the generator \( G_\theta \) and discriminator \( D_\varphi \) feature maps.

C. OPTIMALITY OF SIMULTANEOUS GAME

The minimax optimization in (2) refers to a two-player minimax game. The generator \( G_\theta \) tries to minimize \( S_{c_{\kappa}}^{\varepsilon}(\mu_\theta, \nu) \) with respect to the decision variable \( \theta \). The discriminator \( D_\varphi \) tries to maximize it with respect to \( \varphi \). Neither player knows anything about the critical point of \( S_{c_{\kappa}}^{\varepsilon}(\mu_\theta, \nu) \). Both players follow the rules of the game and can act simultaneously. The minimax theorem in (3) guarantees that the order of which player goes first does not matter.

To be concise we rewrite \( S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \). A well-known notion of optimality in this game is that of a local Nash equilibrium.

**Definition III-C.1.** The point \((\theta^*, \varphi^*)\) is a local Nash equilibrium of \( S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \) if there exists \( \delta > 0 \) such that, for any \( \theta, \varphi \) satisfying \( \|\theta - \theta^*\| \leq \delta \), we have

\[
S_{c_{\kappa}}^{\varepsilon}(\theta^*, \varphi^*) \leq S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \leq S_{c_{\kappa}}^{\varepsilon}(\theta^*, \varphi^*) \leq S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi^*).
\]

(27)

The local Nash equilibrium can also be characterized in terms of the 1st and 2nd order conditions.

**Proposition III-C.1.** If \( S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \) has well-defined gradient and Hessian, then any local Nash equilibrium satisfies the following conditions:

- 1st order: \( \nabla_\theta S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) = 0 \) and \( \nabla_\varphi S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) = 0 \).
- 2nd order: \( \nabla_\theta^2 S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \leq 0 \) and \( \nabla_\varphi^2 S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \geq 0 \).

We note that there exists a twice-differentiable function \( S_{c_{\kappa}}^{\varepsilon}(\theta, \varphi) \) for which pure strategy local Nash equilibria do not exist.

Theoretically, SGD can find an \( \varepsilon \)-approximate stationary point within \( O(\kappa^2 \log(1/\varepsilon)) \) iterations for strongly-convex and strongly-concave objectives, and within \( O(\varepsilon^{-2}) \) iterations with decaying stepsize for a convex-concave simultaneous game [20], [21]. It is hard to analyse the convergence rate for the Sinkhorn divergence. The first theoretical approximation, based on the EOT dual formulation, provides a convergence guarantee to \( \varepsilon \)-stationary in \( O(\sqrt{\varepsilon}/\lambda) \) for SWGAN [8]. This guarantee, however, requires an optimal discriminator.

To empirically determine the SimSGDA Algorithm’s convergence rate, we need a proper metric that measures similarity among the tensors produced by the optimizing variables \( \theta \) and \( \varphi \). We assume that the algorithm can reach a local Nash
Table 2: The implemented DCGAN Neural Architecture.

| Generator Architecture |
|-------------------------|
| ID Image $x \in \mathbb{R}^{32 \times 32 \times 1}$ |
| $4 \times 4$ Conv, 64 stride: $2 \times 2$ + SN + LReLU |
| $4 \times 4$ Conv, 128 stride: $2 \times 2$ + SN + LReLU |
| $4 \times 4$ Conv, 256 stride: $2 \times 2$ + SN + LReLU |
| $4 \times 4$ Conv, 100 stride: $1 \times 1$ + SN |

| Generator Architecture |
|-------------------------|
| $z \in \mathbb{R}^{100} \sim N(0,1)$ |
| $4 \times 4$ TranspConv, 256 + BN + ReLU |
| $4 \times 4$ TranspConv, 128 + BN + ReLU |
| $4 \times 4$ TranspConv, 64 + BN + ReLU |
| $4 \times 4$ TranspConv, 1 + tanh |

Equilibrium if the requisite small quantities in Definition III-C.1 exist. Our work uses the structural similarity index measure (SSIM) from [33]. It quantifies structural information changes, e.g., the local mean, variance, and correlation, over the optimizing generator weight parameter $\theta$, relative to the preceding SimSGDA iteration. By Definition III-C.1, the algorithm achieves a local minmax if there is a small $\delta > 0$ such that $\|\theta - \theta^*\| \leq \delta$ and $\|\varphi - \varphi^*\| \leq \delta$, for any $(\theta, \varphi)$. A more precise treatment is already given in [13].

IV. EXPERIMENTS
We carried out experiments to evaluate our model's convergence and stability. We used the standard DCGAN architecture, like the one used by all prior models, with additional spectral normalization (SN) in the discriminator and a batch normalization (BN) in the generator. The detail is in Table 2. The modifications are designed to significantly smooth the SimSGDA optimization landscape. It is known that BN smooths the objective function [34] while SN enforces Lipschitz continuity [4]. NVIDIA Tesla V-100 GPUs, with 32 GB VRAM each, constitute the platform. The linear-time Sinkhorn Algorithm with Gaussian kernel, AutoDiff, and ADAM optimizer were deployed in all of the experiments. For reproducibility, our PyTorch codes are available online at https://github.com/muchlisinadi/compare-ipmgan.

A. DATASETS SELECTION
Image datasets such as MNIST, CIFAR10, ImageNet, and CelebA have been used to test prior Sinkhorn GAN models. CIFAR10 and CelebA are well-studied datasets of $32 \times 32$ color images. For the experiments, we selected Cats and CelebA, instead of CIFAR10 and ImageNet, for three main reasons. First, the preferred two datasets have around 20,000 images each. This number is much larger than the 600 and 6,000 per class, respectively, in ImageNet and CIFAR10. Second, both samples for Cats and CelebA generated by GANs are easier to inspect visually, compared to multi-class datasets. Finally, using datasets of the same size allows us to focus on the impacts of the hyperparameters in our Sinkhorn GAN and on the visual quality comparison of the generated samples against those produced by other state-of-the-art models.

B. HYPERPARAMETERS SELECTION
The hyperparameters include the mini batch size $m$, the SimSGDA learning rate $\eta$, the number of Sinkhorn iterations $L$, and the entropy regularization weight $\varepsilon$. To determine the best values for the set $\{m, \eta, L, \varepsilon\}$ we performed simulations with varied $\{\eta, \varepsilon\}$ on fixed $\{m, L\}$. We decided on $L = 10$ and $m = 100$ based on the insights gained from the reported experiments in [13].

The outcomes for $\varepsilon \in \{0.001, 0.01, 0.1\}$ and $\eta \in \{5 \cdot 10^{-5}, 10^{-4}\}$ are displayed in Figures 1 and 2. The choice of $\varepsilon$ is crucial. Too much regularization, that is too large an $\varepsilon$, leads to a loose fit on the data. Insufficient regularization severely slows the convergence of the Sinkhorn iteration. It demands more GPU time whenever the total Sinkhorn iteration budget $L$ increases. Choosing $\eta = 5 \cdot 10^{-5}$ produces better generator and discriminator losses compared with the higher $\eta = 10^{-4}$. The latter leads to vanishing gradients. Since $m$ and $L$ are fixed, the best hyperparameters for our model are

$$m = 100, \quad L = 10, \quad \varepsilon = 10^{-3}, \quad \eta = 5 \cdot 10^{-5}.$$ (28)

Choosing lower values for $\varepsilon$, $\eta$, and $L$ produces better quality samples, as illustrated in Figure 4 for Cats. Higher inception score (IS) and lower Fréchet inception distance (FID) scores shown, respectively, in Figures 7 and 8, strengthen our claim of improved sample quality. We trained our model on CelebA with the same hyperparameters. It yielded samples of comparable quality when the positive features are in use, as shown in Figure 5. For higher $\varepsilon$ and $\eta$, SimSGDA tends to diverge above $275 \cdot 10^3$ iterations, after having previously converged. This tendency is highlighted in Figure 1 for $\varepsilon = 0.01$ and in Figure 2 for $\eta = 10^{-4}$. This phenomenon, known as a rotational behaviour, is typical when a constant is chosen as the learning rate $\eta$ [35], [36].

C. IMPACT OF POSITIVE FEATURES
We investigated the impact of using Sinkhorn with positive features on the selected hyperparameters. We ran the SimSGDA Algorithm on two types of Sinkhorn Algorithms, namely, the kernel-first positive features in Algorithm 1 and the original Sinkhorn matrix scaling from [13]. On the same $L$, the former runs in $O(L r n)$ time, underlying the impact of the positive features. The latter takes $O(L n^2)$ time. Figure 3 shows a faster convergence behaviour for Sinkhorn with positive features below $275 \cdot 10^3$ iterations. Soon after, it diverges, exhibiting a rotational behaviour. The generated Cats samples, however, are comparable in quality to those produced by WGAN. This can be seen in Figure 4 for Cats. A direct comparison of the generated samples from the two types of Sinkhorn Algorithm for CelebA is given in Figure 5. The tendency of Sinkhorn with positive features to generate blurry images is confirmed by the slightly lower IS and higher FID values in Figure 6. Similarly blurry CelebA samples are also produced by other Sinkhorn GANs, notably

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Figure 1: The generator and discriminator losses for the indicated regularization parameters.

Figure 2: The generator and discriminator losses for the indicated learning rates.

Figure 3: A comparison of the generator and discriminator losses for Cats: Sinkhorn with and without positive features.
Figure 4: Generated Cats images.

(a) Sinkhorn with Positive Features  (b) SN-GAN  (c) WGAN-GP

Figure 5: Generated CelebA images on Sinkhorn with and without positive features.

(a) Sinkhorn with Positive Features  (b) Sinkhorn only

Figure 6: A comparison of the IS and FID values on CelebA images: Sinkhorn with and without positive features.

(a) Inception Score (IS)  (b) Fréchet Inception Distance (FID)
SiNG and SiPF. These last two models, however, use different objective functions and optimization techniques.

To empirically establish that SimSGDA converges locally to a Nash equilibrium, we reversed the order of minimization and maximization. The SimSGDA, trained with the reversed order, obtains the same results. This forms a strong empirical evidence that SimSGDA has a CC objective, since it satisfies the von Neumann condition in (3). It has already been proven theoretically that SimSGDA with a CC objective can converge to a local Nash equilibrium [22]–[24]. Here we supply empirical evidences for the Sinkhorn GAN trained using SimSGDA. All previous Sinkhorn GANs also have CC objectives, as mentioned in Table 1. They, however, are trained with SeqSGDA, with additional critic updates on the discriminator, not with SimSGDA.

Many methods have been proposed to handle the rotational problem that may appear, as in our case here, in the generator and discriminator losses after convergence. Prominent examples include the 1st order consensus optimization from [35] and the 2nd order follow-the-ridge (FR) correction proposed in [36]. Since the Sinkhorn divergence has well-defined gradient and Hessian, through its dual and semi-dual definitions, both corrections can be implemented. A comprehensive analysis on such implementations deserves a treatment beyond what we can include in the present work.

D. INCEPTION Score AND Fréchet INCEPTION DISTANCE Evaluation metrics

We use two common GAN evaluation metrics, namely the inception score (IS) and the Fréchet inception distance (FID), to evaluate the quality of the generated samples. A high IS score indicates that the generated samples contain clear objects and the GANs can output highly diverse samples. The FID measures the distance of two probability distributions under the assumption that they are Gaussian. It captures the similarity of the generated samples to the real samples better than IS does. A smaller FID value says that the generated samples are more similar to the real ones. High IS and low FID values are desirable.

We used Cats, a DCGAN neural architecture, and the ADAM optimizer, with fixed $(m, L) = (100, 10)$. Figures 7 and 8 show, respectively, the IS and FID values over variations in the regularization $\varepsilon$ and learning rate $\eta$. Our model produces better quality samples as $\varepsilon$ decreases, as indicated by the progressively higher IS and lower FID values. In theory, the Sinkhorn algorithm requires $\varepsilon \to 0$ and $L \to +\infty$ [5]. A clear benefit of using the Sinkhorn iteration in Algorithm 1 is a reduction in the running time, since we do not have to set a high value for $L$.

E. IMPACT OF SINKHORN ITERATION

We have also investigated the effects of the number of iterations on the training stability and the quality of the generated samples. From a numerical point of view, our proposed Sinkhorn algorithm is a fixed-point iteration. Hence, $L$ can be chosen to be large, say $L = 100$, or small, say $L = 10$. Larger $L$ produces images with better quality. One must keep in mind, however, that $L$ also depends on the chosen value of the regularization parameter $\varepsilon$. The latter must be kept small, as was already explained.

We used Cats, a DCGAN neural architecture, and the ADAM optimizer, with fixed $(m, \varepsilon) = (100, 0.001)$. Figure 9 presents the IS and FID charts as $L$ changes. The higher IS and lower FID values indicate that our model produces better quality samples as $L$ decreases. In addition to the theoretical requirement of $L \to +\infty$, the output of the Sinkhorn algorithm also depends on $\varepsilon$ and $m$. Our simulations confirm that $L$ can be kept relatively small in practice.

Intriguingly, some values of $L$, for example $L = 50$ in Figure 10, lead to a vanishing gradient. We therefore hypothesize a trade-off among $m$, $L$, and $\varepsilon$ as the main cause. We note that the Sinkhorn GANs in [7] and [12] use higher values of $m$ and $L$ when using a low $\varepsilon$.

F. COMPARISON TO WGAN

For further evaluation, we benchmark our model against the state-of-the-art WGAN-GP [2] and SN-GAN [4], using Cats and the same DCGAN architecture. We select WGAN-GP and SN-GAN because both are formulated by using the earth-mover distance, which is a variant of OT-based distance. We want to test if our model can generate samples with comparable quality.

A WGAN uses a sequential, instead of simultaneous, SGDA with several critic updates on the discriminator $N_c$. Our model uses the hyperparameters in (28). For WGAN-GP and SN-GAN, we choose $(N_c, m, \eta) = (5, 100, 10^{-4})$. The coefficient for the WGAN-GP’s gradient penalty is fixed to $\lambda = 10$. Figure 4 shows that all three models generate samples of the same quality. Furthermore, the outcomes, plotted for only the generator iterations in Figure 11, confirm the IS and FID convergence in all three models. They follow a similar trend. Compared to ours, WGAN-GP and SN-GAN achieve slightly better sample qualities, indicated by their higher IS and lower FID values. Figure 11 displays how WGAN-GP and SN-GAN achieve convergence faster than ours. Game theoretically, a model may converge faster but not to a local Nash equilibrium or a minimax point [15].

Let $\Delta$ be the total number of gradient updates for both the critics and the discriminators. To approximate the gradient complexity, we performed a linear regression comparison on the captured data with respect to $\log(1/\text{SSIM})$ and $\log(\Delta)$. We made use of the back-and-forward tracks, each with 1,000 iterations, from a stationary point when the generator loss had already converged. The generator parameters are taken at specific gradient updates. Thus, the generator is structurally steady, producing the same tensor. Empirically, the convergence rate of Sinkhorn with positive features is comparable to that of WGAN-GP, as indicated by the positive slopes in Figure 12. Previous study showed that the iteration complexity to return to an $\varepsilon$-stationary can be approximated to $O(\kappa \log(\varepsilon^{-1}))$, where $\kappa$ is a value that depends on the
Figure 7: The IS and FID values for the indicated regularization parameters.

Figure 8: The IS and FID values for the listed learning rates.

Figure 9: The IS and FID values for the given numbers of Sinkhorn iterations.
Sinkhorn divergence’s convexity and the learning rate in SeqSGDA algorithm [13].

V. CONCLUDING REMARKS

We have proposed a new variant of Sinkhorn GAN. The generator is trained to minimize the asymmetric distance $W^c_\varepsilon(\theta, \phi)$ while the discriminator maximizes the symmetric divergence $S^c_\varepsilon(\theta, \phi)$. To accelerate the training, we incorporate a linear-time Sinkhorn with positive features and the more efficient SimSGDA algorithm. The model is shown to be uniquely stable when trained with the right choice for the set of hyperparameters $\{m, L, \eta, \varepsilon\}$ on datasets of images, each of size $32 \times 32$. A detailed comparison between our model and prior Sinkhorn GANs is given in Table 1.

Our theoretical review assumes the existence of a local Nash equilibrium when using the SimSGDA Algorithm with CC objective. An investigation to explain this fact is an open direction to pursue. We have highlighted that our method works well in practice. One downside of SimSGDA in deployment is that it runs into the notorious rotational behavior at higher iterations when a low $\varepsilon$ and a fixed $\eta$ are chosen. We also discover that the method becomes less stable in deeper architectures such as ResNet. This may be because the gradients in such architectures can vary greatly in scale. We are currently looking into the observed rotational and
architectural issues. Another task is to scale the approach up to yield higher resolution images.

Our model holds a two-fold advantage over prior Sinkhorn GANs. First, the time complexity $O(n^r)$ in approximating the Sinkhorn divergence $S^r_c(\theta, \varphi)$ is linear. This allows for a significantly enlarged batch sizes. Second, the approach is fully differentiable. It directly computes the gradients of $V^r_c(\theta, \varphi)$ and $S^r_c(\theta, \varphi)$ with respect to $\theta$ and $\varphi$. In comparison, OTGAN does not differentiate directly through $S^r_c(\theta, \varphi)$. It uses, instead, a mini-batch energy distance. SWGAN requires an optimal discriminator oracle whereas SGD-AutoDiff differentiates through the iterations of the Sinkhorn Algorithm. It must, therefore, keep track of the ground cost computation and is applicable only when the entropy regularization $\varepsilon$ is large, since the number of Sinkhorn iteration $L$ cannot be too large.

We have demonstrated that the more efficient SimSGDA can be used to train Sinkhorn GAN. We show that a minimax optimization with a convex-concave Sinkhorn objective problem can be solved to a local Nash equilibrium, under some mild approximations. The overall insight is that the approximations become simpler and computationally more efficient than prior Sinkhorn GANs, while still generating images with comparable quality to state-of-the-art WGAN-GP and SN-GAN. As we continue to strengthen the theoretical foundation, we hope that our model can inspire a wider deployment of EOT in GANs. Our results open up possibilities for future research. For wider practical implementation, the rotational issue and deeper neural architecture adaptation are open for exploration. On the theoretical side, the proof of local Nash equilibrium, the generalization properties as well as the analytical convergence rate of SimSGDA Algorithm in our proposed Sinkhorn GAN are worth investigating.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. ICMIL 2017, Sydney, NSW, Australia, PMLR vol. 70, pp. 214–223, 2017.
[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in Proc. NIPS 2017, Long Beach, CA, USA, pp. 5767–5777, 2017.
[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. NIPS 2014, Montréal, Quebec, Canada, pp. 2672–2680, 2014.
[4] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshidah, “Spectral normalization for generative adversarial networks,” in Proc. ICLR 2018, Vancouver, BC, Canada.
[5] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in Proc. NIPS 2013, Lake Tahoe, Nevada, USA, pp. 2292–2300, 2013.
[6] A. Genevay, G. Peyré, and M. Cuturi, “Learning generative models with Sinkhorn divergences,” in Proc. AISTATS 2018, Playa Blanca, Canary Islands, Spain, pp. 1608–1617, 2018.
[7] T. Salimans, H. Zhang, A. Radford, and D. N. Metaxas, “Improving GANs using optimal transport,” poster presentation in ICLR 2018, Vancouver, BC, Canada.
[8] M. Sanjabi, J. Ba, M. Razaviyayn, and J. D. Lee, “On the convergence and robustness of training GANs with regularized optimal transport,” in Proc. NeurIPS 2018, Montréal, Québec, Canada, pp. 7091–7101, 2018.
[9] D. Liu, M. T. Yu, S. Chatterjee, and L. K. Rasmussen, “Entropy-regularized optimistic transport generative models,” in Proc. ICASSP 2019, Brighton, UK, pp. 3532–3536, 2019.
[10] A. Genevay, L. Chizat, F. R. Bach, M. Cuturi, and G. Peyré, “Sample complexity of Sinkhorn divergences,” in Proc. AISTATS 2019, Naha, Okinawa, Japan, pp. 1574–1583, 2019.
[11] F. Paty and M. Cuturi, “Regularized optimal transport is ground cost adversarial,” in Proc. ICML 2020, virtual event, PMLR vol. 119, pp. 7532–7542, 2020.
[12] M. Scetbon and M. Cuturi, “Linear time Sinkhorn divergences using positive features,” in Proc. NeurIPS 2020, virtual event, pp. 13468–13480, 2020.
[13] R. Adnan, M. A. Saputra, J. Fadil, M. F. Ezerman, M. Iqbal, and T. Basaruddin, “Convergence of non-convex non-concave GANs using Sinkhorn divergence,” IEEE Access, vol. 9, pp. 67595–67609, 2021.
[14] A. Genevay, “Entropy-Regularized Optimal Transport for Machine Learning,” Ph.D. dissertation, Univ. Paris Sciences et Lettres, Mar. 2019.
[15] C. Jin, P. Netrapalli, and M. I. Jordan, “What is local optimality in nonconvex-nonconcave minimax optimization?” in Proc. ICML 2020, virtual event, PMLR vol. 119, pp. 4880–4889, 2020.
[16] Sebastien Bubeck, “Convex optimization: Algorithms and complexity,” Found. Trends Mach. Learn., vol. 8 no. 3–4, pp. 231–357, 2015.
[17] J. von Neumann, “Zur theorie der gesellschaftsspiele,” Annalen der Mathematik und der Physik, vol. 8 no. 3–4, pp. 231–357, 2015.
[18] F. A. Osiadzisz, “Strategic behavior of customers in an economic model,” Naha, Okinawa, Japan, PMLR vol. 89, pp. 907–915, 2019.
[19] F. Farnia and A. E. Ozdaglar, “Train simultaneously, generalize better: Stability of gradient-based minimax learners,” [Online] Available: arxiv.org/pdf/2010.12561.
[20] F. Shi, Z. Wang, A. Ribeiro, and H. Hassani, “Sinkhorn natural gradient for generative models,” in Proc. NeurIPS 2020, virtual event, pp. 1646–1656, 2020.
[21] M. Cuturi and G. Peyré, “Semidual regularization of optimal transport,” SIAM Rev., vol. 60, no. 4, pp. 941–965, Jan. 2018.
[22] T. Lin, C. Jin, and M. I. Jordan, “On gradient descent ascent for nonconvex-concave minimax problems,” in Proc. ICML 2019, Naha, Okinawa, Japan, PMLR vol. 89, pp. 486–495, 2019.
[23] T. Liang and J. Stokes, “Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks,” in Proc. AISTATS 2019, Naha, Okinawa, Japan, PMLR vol. 89, pp. 907–915, 2019.
[24] S. S. Du and W. Hu, “Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity,” in Proc. AISTATS 2019, Naha, Okinawa, Japan, PMLR vol. 89, pp. 196–205, 2019.
[25] A. Nemirovski, “Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems,” SIAM J. Optim., vol. 15, no. 1, pp. 229–251, Jan. 2004.
[26] A. Nedić and A. E. Ozdaglar, “Subgradient methods for saddle-point problems,” J. Optim. Theory Appl., vol. 142, no. 1, pp. 205–228, 2009.
[27] G. Peyré and M. Cuturi, “Computational optimal transport: With applications to data science,” Found. Trends Mach. Learn., vol. 11, no. 5–6, pp. 355–607, 2019.
[28] C. Villani, Optimal Transport: Old and New, Grundr¨uben der mathematischen Wissenschaften. Berlin Heidelberg, Germany: Springer, 2008.
[29] L. Chizat, “Unbalanced Optimal Transport: Models, Numerical Methods, Applications,” Ph.D. dissertation, Univ. Paris Sciences et Lettres, Nov. 2017.
[30] A. Genevay, “Entropy-Regularized Optimal Transport for Machine Learning,” Grundlehren der mathematischen Wissenschaften. Berlin Heidelberg, Germany: Springer, 2008.
[31] S. S. Du, Z. Wang, A. Ribeiro, and H. Hassani, “Sinkhorn natural gradient for generative models,” in Proc. NeurIPS 2020, virtual event, pp. 1646–1656, 2020.