
Towards a Deeper Understanding of Adversarial Losses

Hao-Wen Dong¹ Yi-Hsuan Yang¹

Abstract

Recent work has proposed various adversarial losses for training generative adversarial networks. Yet, it remains unclear what certain types of functions are valid adversarial loss functions, and how these loss functions perform against one another. In this paper, we aim to gain a deeper understanding of adversarial losses by decoupling the effects of their component functions and regularization terms. We first derive some necessary and sufficient conditions of the component functions such that the adversarial loss is a divergence-like measure between the data and the model distributions. In order to systematically compare different adversarial losses, we then propose DANTest—a new, simple framework based on discriminative adversarial networks. With this framework, we evaluate an extensive set of adversarial losses by combining different component functions and regularization approaches. This study leads to some new insights into the adversarial losses. For reproducibility, all source code is available at <https://github.com/salu133445/dan>.

1. Introduction

Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a class of unsupervised machine learning algorithms. In essence, a GAN learn a generative model with the guidance of another discriminative model which is trained jointly. However, the idea of adversarial losses is not limited to unsupervised learning. Adversarial losses can also be applied to supervised and semi-supervised scenarios (e.g., (Isola et al., 2017; dos Santos et al., 2017)). Over the past few years, adversarial losses have advanced the state of the art in many fields (Goodfellow, 2016).

Despite the success, there are several open questions that need to be addressed. On one hand, although plenty adversarial losses have been proposed, we have little theoretical understanding of what makes a loss function a valid one.

On the other hand, we note that any two adversarial losses can differ in terms of not only the *component functions* (e.g., minimax or hinge; see Section 2) used in the main loss function that sets up the two-player adversarial game, but also the *regularization approaches* (e.g., gradient penalties (Gulrajani et al., 2017)) used to regularize the models. However, it remains unclear how they respectively contribute to the performance of an adversarial loss. In other words, when empirically compare two adversarial losses, we need to decouple the effects of the component functions and the regularization terms, otherwise we cannot tell which one of them makes an adversarial loss better than the other.

Among existing comparative analysis of adversarial losses, to the best of our knowledge, only Lucic et al. (2018) and Kurach et al. (2018) attempted to decouple the effects of the component functions and regularization approaches. But, only few combinations of component functions and regularization approaches were tested in these two prior works, only seven and nine respectively. We attribute this to the high computational cost that may involve to conduct the experiments, and, more importantly, the lack of a framework to systematically evaluate adversarial losses.

These two research questions can be summarized as follows:

- RQ1 What certain types of component functions are theoretically valid adversarial loss functions?
- RQ2 How different combinations of the component functions and the regularization approaches perform empirically against one another?

We aim to tackle these two RQs in this paper to advance our understanding of the adversarial losses. Specifically, our contribution to RQ1 is based on the intuition that a favorable adversarial loss should be a divergence-like measure between the distribution of the real data and the distribution of the model output, since in this way we can use the adversarial loss as the training criterion to learn the model parameters. We derive necessary and sufficient conditions such that an adversarial loss has such a favorable property (Sections 3.3 and 3.4). Interestingly, our theoretical analysis leads to a new perspective to understand the underlying game dynamics of adversarial losses (Section 3.6).

For RQ2, we need an efficient way to compare different adversarial losses. Hence, we adopt the discriminative adver-

¹Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan. Correspondence to: Hao-Wen Dong <salu133445@citi.sinica.edu.tw>, Yi-Hsuan Yang <yang@citi.sinica.edu.tw>.

	f	g	h	y^*
minimax (Goodfellow et al., 2014)	$-\log(1 + e^{-y})$	$-y - \log(1 + e^{-y})$	$-y - \log(1 + e^{-y})$	0
nonsaturating (Goodfellow et al., 2014)	$-\log(1 + e^{-y})$	$-y - \log(1 + e^{-y})$	$\log(1 + e^{-y})$	0
Wasserstein (Arjovsky et al., 2017)	y	$-y$	$-y$	0
least squares (Mao et al., 2017)	$-(y - 1)^2$	$-y^2$	$(y - 1)^2$	$\frac{1}{2}$
hinge (Lim & Ye, 2017; Tran et al., 2017)	$\min(0, y - 1)$	$\min(0, -y - 1)$	$-y$	0

Table 1. Component functions for a few adversarial losses (see (1) and (2)). y^* denotes the root of $f(y) = g(y)$ and $f'(y) = -g'(y)$.

sarial networks (DANs) (Mirza & Osindero, 2014), which are essentially conditional GANs with both the generator and the discriminator being discriminative models. Based on DANs, we propose *DANTest*—a new, simple framework for comparing adversarial losses (Section 4). The main idea is to first train a number of DANs for a supervised learning task (e.g., classification) using different adversarial losses, and then compare their performance using standard evaluation metrics for supervised learning (e.g., classification accuracy). With the DANTest, we systematically evaluate 168 adversarial losses featuring the combination of ten existing component functions, two new component functions we originally propose in this paper in light of our theoretical analysis, and 14 existing regularization approaches (Section 5). Moreover, we use the DANTest to empirically study the effect of the Lipschitz constant (Arjovsky et al., 2017), penalty weights (Mescheder et al., 2018), momentum terms (Kingma & Ba, 2014), and others. We discuss the new insights that are gained, and their implications to the design of adversarial losses in future research.

2. Background

2.1. Generative Adversarial Networks

A generative adversarial network (Goodfellow et al., 2014) is a generative latent variable model that aims to learn a mapping from a latent space \mathcal{Z} to the data space \mathcal{X} , i.e., a generative model G , which we will refer to as the *generator*. A discriminative model D (i.e., the *discriminator*) defined on \mathcal{X} is trained alongside the G to provide guidance for it. Let p_d denote the *data distribution* and p_g be the *model distribution* implicitly defined by $G(\mathbf{z})$ when $\mathbf{z} \sim p_z$. In general, most GAN loss functions proposed in the literature can be formulated as:

$$\max_D \mathbb{E}_{\mathbf{x} \sim p_d} [f(D(\mathbf{x}))] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p_g} [g(D(\tilde{\mathbf{x}}))], \quad (1)$$

$$\min_G \mathbb{E}_{\tilde{\mathbf{x}} \sim p_g} [h(D(\tilde{\mathbf{x}}))], \quad (2)$$

where f, g and h are real functions defined on the data space (i.e., $\mathcal{X} \rightarrow \mathbb{R}$) and we will refer to them as the *component functions*. We summarize in Table 1 the component functions f, g and h used in some existing adversarial losses.

Some prior work has also investigated the so-called IPM-

based GANs, where the discriminator is trained to estimate an integral probability metric (IPM) between p_d and p_g :

$$d(p_d, p_g) = - \sup_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{x} \sim p_d} [D(\mathbf{x})] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p_g} [D(\tilde{\mathbf{x}})], \quad (3)$$

where \mathcal{D} is a set of functions from \mathcal{X} to \mathbb{R} . For example, the Wasserstein GANs (Arjovsky et al., 2017) consider \mathcal{D} to be the set of all 1-Lipschitz functions. Other examples include McGAN (Mroueh et al., 2017), MMD GAN (Li et al., 2017) and Fisher GAN (Mroueh & Sercu, 2017). Please note that the main difference between (1) and (3) is that in the latter we constrain D to be in some set of functions \mathcal{D} .

2.2. Gradient Penalties

As the discriminator is often found to be too strong to provide reliable gradients to the generator, one regularization approach is to use some gradient penalties to constrain the modeling capability of the discriminator. Most gradient penalties proposed in the literature take the following form:

$$\lambda \mathbb{E}_{\tilde{\mathbf{x}} \sim p_{\tilde{\mathbf{x}}}} [R(\|\nabla_{\tilde{\mathbf{x}}} D(\tilde{\mathbf{x}})\|)], \quad (4)$$

where the *penalty weight* $\lambda \in \mathbb{R}$ is a pre-defined constant, and $R(\cdot)$ is a real function. The distribution $p_{\tilde{\mathbf{x}}}$ defines where the gradient penalties are enforced. Table 2 shows the distribution $p_{\tilde{\mathbf{x}}}$ and function R used in some common gradient penalties. And, Figure 1 illustrates $p_{\tilde{\mathbf{x}}}$.

When gradient penalties are enforced, the loss function for training the discriminator contains not only the component functions f and g in (1) but also the *regularization term* (4).

2.3. Spectral Normalization

Another regularization approach we consider is the spectral normalization proposed by Miyato et al. (2018). It normalizes the spectral norm of each layer in a neural network to enforce the Lipschitz constraints. While the gradient penalties introduced in Section 2.2 impose local regularizations, the spectral normalization imposes a global regularization on the discriminator. Therefore, it is possible to combine the spectral normalization with the gradient penalties. We will examine this in Section 5.3.

	$p_{\tilde{\mathbf{x}}}$	$R(x)$
coupled gradient penalties (Gulrajani et al., 2017)	$p_d + U[0, 1] (p_g - p_d)$	$(x - k)^2$ or $\max(x, k)$
local gradient penalties (Kodali et al., 2017)	$p_d + c N(0, I)$	$(x - k)^2$ or $\max(x, k)$
R ₁ gradient penalties (Mescheder et al., 2018)	p_d	x
R ₂ gradient penalties (Mescheder et al., 2018)	p_g	x

Table 2. Distribution $p_{\tilde{\mathbf{x}}}$ and function R in (4) for common gradient penalties, where $c, k \in \mathbb{R}$ are considered hyperparameters (k is the Lipschitz constant). We will refer to the $(x - k)^2$ and the $\max(x - k)$ versions as the *two-side* and the *one-side* penalties, respectively.

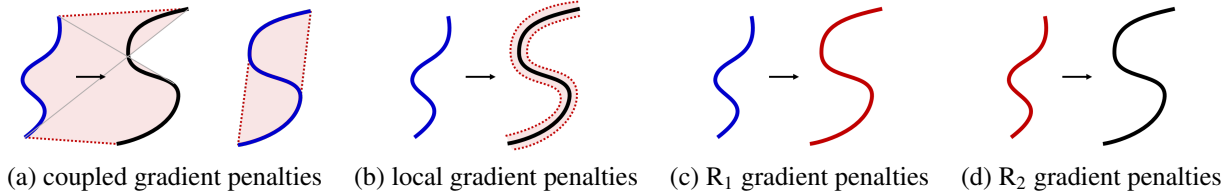


Figure 1. Illustrations of the regions in the data space where gradient penalties are imposed (i.e., the support of $p_{\tilde{\mathbf{x}}}$) for common gradient penalties, shown as the red shaded area in (a) and (b) and the red curves in (c) and (d). The blue and black curves denote the model and the data manifolds, respectively. The right figure in (a) shows the case when the generator perfectly fabricates the data distribution (i.e., $p_g = p_d$). For (c) and (d), the gradient penalties are enforced directly on the model and the data manifolds, respectively.

3. Theoretical Results

In the following analysis, we follow the notations in (1) and (2). Proofs can be found in Appendix A.

3.1. Favorable properties for adversarial losses

Let us first consider the minimax formulation:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_d} [f(D(\mathbf{x}))] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p_g} [g(D(\tilde{\mathbf{x}}))]. \quad (5)$$

We can see that if the discriminator is able to reach optimality, the training criterion for the generator is

$$L_G = \max_D \mathbb{E}_{\mathbf{x} \sim p_d} [f(D(\mathbf{x}))] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p_g} [g(D(\tilde{\mathbf{x}}))]. \quad (6)$$

In general, for a valid adversarial loss, the discriminator is responsible for providing a measure of the discrepancy between the data distribution p_d and the model distribution p_g . In principle, this will then serve as the training criterion for the generator to push p_g towards p_d . Hence, we would like such an adversarial loss to be a divergence-like measure between p_g and p_d . From this view, we can now define the following two favorable properties of adversarial losses.

Property 1. (Weak favorable property) *For any fixed p_d , L_G has a global minimum at $p_g = p_d$.*

Property 2. (Strong favorable property) *For any fixed p_d , L_G has a unique global minimum at $p_g = p_d$.*

We can see that Property 2 makes $L_G - L_G^*$ a divergence of p_d and p_g for any fixed p_d , where $L_G^* = L_G|_{p_g=p_d}$, and Property 1 provides a weaker version when the identity of indiscernibles is not necessary. Note that L_G is not a divergence since $L_G \geq 0$ does not always hold.

3.2. Ψ and ψ functions

In order to derive some necessary and sufficient conditions for Properties 1 and 2, we first observe from (6) that

$$L_G = \max_D \int_{\mathbf{x}} p_d(\mathbf{x}) f(D(\mathbf{x})) + p_g(\mathbf{x}) g(D(\mathbf{x})) d\mathbf{x} \quad (7)$$

$$= \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \max_D \left(\frac{p_d(\mathbf{x}) f(D(\mathbf{x}))}{p_d(\mathbf{x}) + p_g(\mathbf{x})} + \frac{p_g(\mathbf{x}) g(D(\mathbf{x}))}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \right) d\mathbf{x}. \quad (8)$$

Now, if we let $\tilde{\gamma} = \frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})}$ and $\tilde{y} = D(\mathbf{x})$, we get

$$L_G = \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \max_{\tilde{y}} \tilde{\gamma} f(\tilde{y}) + (1 - \tilde{\gamma}) g(\tilde{y}) d\mathbf{x}. \quad (9)$$

Please note that $\tilde{\gamma}(\mathbf{x}) = \frac{1}{2}$ if and only if $p_d(\mathbf{x}) = p_g(\mathbf{x})$. Let us now consider the terms inside the integral and define the following two functions:

$$\Psi(\gamma, y) = \gamma f(y) + (1 - \gamma) g(y), \quad (10)$$

$$\psi(\gamma) = \max_y \Psi(\gamma, y), \quad (11)$$

where $\gamma \in [0, 1]$ and $y \in \mathbb{R}$ are two variables independent of \mathbf{x} . We visualize in Figures 2(a)–(d) the Ψ and ψ functions for different common adversarial losses (see Appendix B for the graphs of the ψ functions alone). These two functions actually reflect some important characteristics of the adversarial losses (see Section 3.6) and will be used intensively in our theoretical analysis.

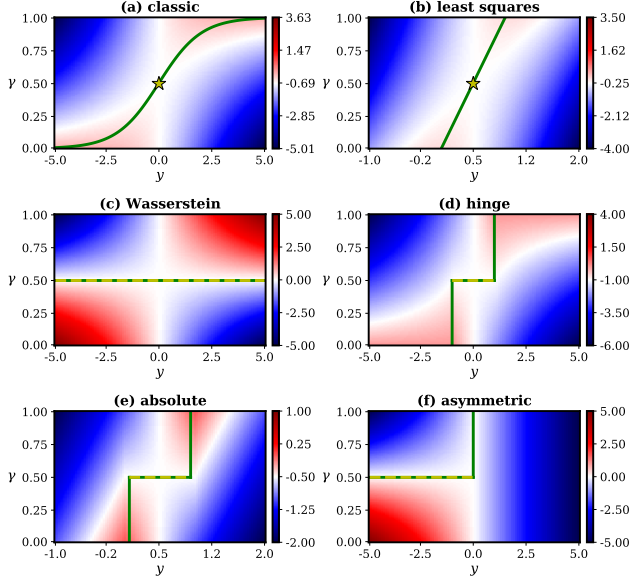


Figure 2. Graphs of the Ψ functions of different adversarial losses. The green lines show the domains of the ψ functions (i.e., the value(s) that y can take for different γ in (11)). The star marks, and any points on the yellow dashed lines, are the minimum points of ψ . The midpoints of the color maps are intentionally set to the minima of ψ (i.e., the values taken at the star marks or the yellow segments). Note that $\gamma \in [0, 1]$ and $y \in \mathbb{R}$, so we plot different portions of y where the characteristics of Ψ can be clearly seen.

3.3. Necessary conditions for the favorable properties

For the necessary conditions of Properties 1 and 2, we have the following two theorems.

Theorem 1. *If Property 1 holds, then for any $\gamma \in [0, 1]$, $\psi(\gamma) + \psi(1 - \gamma) \geq 2\psi(\frac{1}{2})$.*

Theorem 2. *If Property 2 holds, then for any $\gamma \in [0, 1] \setminus \{\frac{1}{2}\}$, $\psi(\gamma) + \psi(1 - \gamma) > 2\psi(\frac{1}{2})$.*

With Theorems 1 and 2, we can easily check if a pair of component functions f and g form a valid adversarial loss.

3.4. Sufficient conditions for the favorable properties

For sufficient conditions, we have two theorems as follows.

Theorem 3. *If $\psi(\gamma)$ has a global minimum at $\gamma = \frac{1}{2}$, then Property 1 holds.*

Theorem 4. *If $\psi(\gamma)$ has a unique global minimum at $\gamma = \frac{1}{2}$, then Property 2 holds.*

We also have the following theorem for a more specific guideline for choosing the component functions f and g .

Theorem 5. *If $f'' + g'' \leq 0$ and there exists some y^* such that $f(y^*) = g(y^*)$ and $f'(y^*) = -g'(y^*) \neq 0$, then $\psi(\gamma)$ has a unique global minimum at $\gamma = \frac{1}{2}$.*

By Theorems 4 and 5, we now see that any component function pair f and g that satisfies the prerequisites in Theorem 5 makes $L_G - L_G^*$ a divergence between p_d and p_g for any fixed p_d . Interestingly, while such a theoretical analysis has not been done before, it happens that all the adversarial loss functions listed in Table 1 have such favorable properties. We intend to examine in Section 5.2 empirically the cases when the prerequisites of Theorem 5 do not hold.

In practice, the discriminator often cannot reach optimality at each iteration. Therefore, as discussed by Nowozin et al. (2016); Fedus et al. (2018), the objective of the generator is similar to variational divergence minimization (i.e., to minimize a lower bound of some divergence between p_d and p_g), where the divergence is estimated by the discriminator.

3.5. Loss functions for the generator

Intuitively, the generator should minimize the divergence-like measure estimated by the discriminator. We have accordingly $h = g$. However, some prior works have investigated setting h different from g . In general, most of these alternative generator losses do not change the solutions of the game and are proposed based on some heuristics. While our theoretical analysis concerns with only f and g , we intend to empirically examine the effects of the generator loss function h in Section 5.4.

3.6. Analyzing the adversarial game by the Ψ functions

Figure 2 gives us some new insights regarding the adversarial behaviors of the discriminator and the generator. On one hand, if we follow (9) and consider $\tilde{y} = D(\mathbf{x})$ and $\tilde{\gamma}(\mathbf{x}) = \frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})}$, then the discriminator can be viewed as maximizing Ψ along the \tilde{y} -axis. On the other hand, since the generator is trained to push p_g towards p_d , it can be viewed as minimizing Ψ along the $\tilde{\gamma}$ -axis. In this way, we can see why all these Ψ functions are saddle-shaped and have saddle points at $\gamma = \frac{1}{2}$ (i.e., when $p_d(\mathbf{x}) = p_g(\mathbf{x})$).

Ideally, if the discriminator can be trained till optimality, then we will be on the green line, the domain of the ψ function. In this case, the generator can be viewed as minimizing Ψ along the green line (i.e., minimizing ψ). Note that as L_G is an integral over all possible \mathbf{x} , such adversarial game is actually being played in a (usually) high dimensional space.

By designing the landscape of Ψ , we propose and consider two new losses in our empirical study in Section 5.3:

- The *absolute* loss, with $f(y) = -h(y) = -|1 - y|$, $g(y) = -|y|$. Its Ψ -landscape is similar to those of the least squares and the hinge losses (see Figure 2(e)).
- The *asymmetric* loss, with $f(y) = -|y|$, $g(y) = h(y) = -y$. Its Ψ -landscape is similar to that of the Wasserstein loss, but the positive part of y is blocked (see Figure 2(f)).

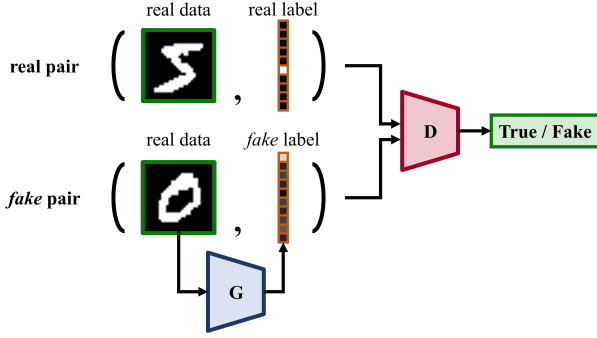


Figure 3. An example of a DAN for digit classification. G is now a discriminative model that aims to predict the label of a real data sample. D takes as input either a “(real data, real label)” or a “(real data, fake label)” pair and aims to examine its authenticity.

4. DANTest

Discriminative adversarial networks (DANs) (dos Santos et al., 2017) are essentially conditional GANs (Mirza & Osindero, 2014) where both the generator and the discriminator are discriminative models, as shown in Figure 3. Based on DANs, we propose a new, simple framework, dubbed *DANTest*, for systematically comparing different adversarial losses. Specifically, the DANTest goes as follows:

- Step 1** Build several DANs. For each of them, the generator G takes as input a real sample and outputs a fake label. The discriminator takes as input a real sample with either its true label, or a fake label made by G , and outputs a scalar indicating if the “sample-label” pair is real.
- Step 2** Train the DANs with different component loss functions, regularization approaches or hyperparameters.
- Step 3** Predict the labels of test data by the trained models.
- Step 4** Compare the performance of different models with standard evaluation metrics used in supervised learning.

Note that the generator is no longer a generative model in this framework, while the discriminator is still trained by the same loss function to measure the discrepancy between p_d and p_g . This way, we can still gain insight into the performance and stability for different adversarial losses. Moreover, although we take a classification task as an example here, the proposed framework is generic and can be applied to other supervised learning tasks as well, as long as the evaluation metrics for that task are well defined.

An extension of the proposed framework is the *imbalanced dataset test*, where we examine the ability of different adversarial losses on datasets that feature class imbalance. This can serve as a measure of the *mode collapse* phenomenon (Che et al., 2017), which is a commonly-encountered failure case in GAN training. By testing on datasets with different levels of imbalance, we can examine how different adversarial losses suffer from the mode collapse problem.

	nonsaturating	Wasserstein	hinge
$\epsilon = 0.5$	8.47 ± 0.36	73.16 ± 6.36	15.20 ± 2.46
$\epsilon = 0.9$	8.96 ± 0.63	57.66 ± 5.13	8.94 ± 0.87
$\epsilon = 1.0$	8.25 ± 0.35	5.89 ± 0.26	6.59 ± 0.31
$\epsilon = 1.1$	8.62 ± 0.45	60.30 ± 7.61	8.02 ± 0.35
$\epsilon = 2.0$	9.18 ± 0.94	69.54 ± 5.37	11.87 ± 0.85

Table 3. Error rates (%) for the ϵ -weighted versions of the non-saturating, the Wasserstein and the hinge losses (see (12)) on the standard dataset. Here, $\epsilon = 1.0$ corresponds to the original losses.

5. Experiments and Results

5.1. Datasets and Implementation Details

All the experiments reported here are done based on the DANTest. If not otherwise specified, we use the MNIST handwritten digits database (LeCun et al., 1998), which we refer to as the **standard** dataset. As it is class-balanced, we create two imbalanced versions of it. The first one, referred to as the **imbalanced** dataset, is created by augmenting the training samples for digit ‘0’ by shifting them each by one pixel to the top, bottom, left and right, so that it contains *five* times more training samples of ‘0’ than the standard dataset. Moreover, we create the **very imbalanced** dataset, where we have *seven* times more training samples for digit ‘0’ than the standard dataset. For other digits, we randomly sample from the standard dataset and intentionally make the sizes of the resulting datasets identical to that of the standard dataset. We use the same test set for all the experiments.

We implement G and D as convolutional neural networks (see Appendix C for the network architectures). We use the batch normalization (Ioffe & Szegedy, 2015) in G . If the spectral normalization is used, we only apply it to D , otherwise we use the layer normalization (Ba et al., 2016) in D . We concatenate the label vector to each layer of D . For the gradient penalties, we use Euclidean norms and set λ to 10.0 (see (4)), k to 1.0 and c to 0.01 (see Table 2). We use the Adam optimizers (Kingma & Ba, 2014) with $\alpha = 0.001$, $\beta_1 = 0.0$ and $\beta_2 = 0.9$. We alternatively update G and D once in each iteration and train the model for 100,000 generator steps. The batch size is 64. We implement the model in Python and TensorFlow (Abadi et al., 2016). We run each experiment for ten runs and report the mean and the standard deviation of the error rates.

5.2. Examining the necessary conditions for favorable adversarial loss functions

As discussed in Section 3.4, we examine here the cases when the prerequisites in Theorem 5 do not hold. We consider the classic nonsaturating, the Wasserstein and the hinge losses and change the training objective for the discriminator into

$$\max_D \epsilon \mathbb{E}_{\mathbf{x} \sim p_d} [f(D(\mathbf{x}))] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p_g} [g(D(\tilde{\mathbf{x}}))], \quad (12)$$

	unregularized	TCGP	TLGP	R ₁ GP	R ₂ GP	SN	SN + TCGP	SN + TLGP	SN + R ₁ GP	SN + R ₂ GP
classic (M) (2014)	9.11±0.63	5.65±0.27	5.42±0.17	19.01±3.73	12.91±1.13	7.37±0.52	5.55±0.37	5.57±0.28	11.16±2.66	14.00±2.49
classic (N) (2014)	26.83±7.17	5.64±0.23	5.56±0.31	14.67±4.86	13.80±3.20	8.25±0.35	5.52±0.16	5.61±0.50	12.98±2.71	13.50±3.78
classic (L)	17.38±5.16	5.66±0.36	5.55±0.16	18.49±5.51	14.92±5.20	7.98±0.36	5.70±0.36	5.48±0.29	15.45±6.54	17.61±7.60
hinge (M)	5.57±0.26	4.83±0.34	4.88±0.25	7.31±1.49	9.49±5.30	6.22±0.23	4.93±0.20	5.06±0.33	10.62±2.10	12.91±4.29
hinge (N)	37.55±20.22	5.00±0.24	4.97±0.24	7.34±1.83	7.54±1.31	6.90±0.33	5.05±0.22	5.06±0.39	11.91±4.02	12.10±4.74
hinge (L) (2017; 2017)	11.50±5.32	5.01±0.26	4.89±0.18	8.96±3.55	7.71±1.82	6.59±0.31	4.97±0.19	5.18±0.27	13.63±4.13	11.35±3.40
Wasserstein (2017)	7.69±0.33	5.04±0.19	4.92±0.23	13.89±20.64	7.25±1.19	5.89±0.26	5.50±0.18	5.76±0.70	13.74±5.47	13.82±4.93
least squares (2017)	7.15±0.47	7.27±0.44	6.70±0.44	30.12±28.43	32.44±21.05	7.88±0.45	6.69±0.25	7.11±0.37	9.91±1.55	11.56±4.09
relativistic (2018)	90.20±0.00	5.25±0.25	5.01±0.31	8.00±1.63	8.75±5.83	7.14±0.39	5.35±0.29	5.25±0.26	9.31±2.01	8.62±0.59
relativistic hinge (2018)	52.01±9.38	8.28±10.26	4.71±0.12	8.39±1.92	7.67±1.82	6.44±0.16	5.02±0.31	5.03±0.21	12.56±4.42	12.40±4.55
absolute	6.69±0.24	5.23±0.29	5.20±0.26	8.01±1.96	6.64±0.51	6.79±0.45	5.23±0.13	5.18±0.35	10.42±3.07	9.93±2.28
asymmetric	7.81±0.27	4.77±0.34	4.94±0.14	8.79±3.18	7.33±1.01	5.98±0.40	5.60±0.29	5.82±0.44	8.46±0.43	8.80±1.18

Table 4. Error rates (%) for different adversarial losses and regularization approaches, on the standard dataset. See Section 5.3 and 5.4 for the abbreviations. Underlined and bold fonts indicate respectively entries with the lowest and lowest-three mean error rates per column.

where $\epsilon \in \mathbb{R}$ is a constant. The prerequisites in Theorem 5 do not hold when $\epsilon \neq 1$. We illustrate the Ψ functions of these ϵ -weighted losses in Appendix B.

Table 3 shows the results for $\epsilon = 0.5, 0.9, 1.0, 1.1, 2.0$, using the spectral normalization for regularization. We can see that all the original losses (i.e., $\epsilon = 1$) result in the lowest error rates. In general, the error rates increase as ϵ goes away from 1.0. Notably, the Wasserstein loss turn out failing with error rates over 50% when $\epsilon \neq 1$.

5.3. On different discriminator loss functions

In this experiment, we aim to compare different discriminator loss functions. Specifically, we evaluate an comprehensive set (in total 168) of different combinations of component functions and regularization approaches.

For the component functions, we consider the classic minimax and the classic nonsaturating losses (Goodfellow et al., 2014), the Wasserstein loss (Arjovsky et al., 2017), the least squares loss (Mao et al., 2017), the hinge loss (Lim & Ye, 2017; Tran et al., 2017), the relativistic average and the relativistic average hinge losses (Jolicoeur-Martineau, 2018), as well as the absolute and the asymmetric losses we propose and describe in Section 3.6.

For the regularization approaches, we consider the coupled, the local, the R₁ and the R₂ gradient penalties (GP) and the spectral normalization (SN). For the coupled and the local gradient penalties, we examine both the two-side and the one-side versions (see Table 2). We will use in the captions **OCGP** and **TCGP** as the shorthands for the one-side and the two-side coupled gradient penalties, respectively, and **OLCP** and **TLCP** for the one-side and the two-side local gradient penalties, respectively. We also consider the combinations of the SN with different gradient penalties.

We report in Table 4 the results for all the combinations and present in Figure 4 the training progress for the nonsaturating and the hinge losses. We can see that *there is no single*

winning component functions and regularization approach across all different settings. Some observations are:

With respect to the **component functions**—

- The classic minimax and nonsaturating losses never get the lowest three error rates for all different settings.
- The hinge, the asymmetric and the two relativistic losses are robust to different regularization approaches and tend to achieve lower error rates.
- The relativistic average loss outperforms both the classic minimax and nonsaturating losses across all regularization approaches. But, the relativistic average hinge loss does not always outperform the standard hinge loss.

With respect to the **regularization approaches**—

- The coupled and the local GPs outperform the R₁ and the R₂ GPs across nearly all different component functions, no matter whether the SN is used or not.
- The coupled and the local GPs stabilize the training (see Figure 4) and tend to have lower error rates.
- The R₂ gradient penalties achieve lower error rates than the R₁ gradient penalties. In some cases, they can be too strong and even stop the training early (see Figure 4 (a)).¹
- Combining either the coupled or the local GP with the SN usually leads to higher error rates than using the coupled or the local GP only.
- Similarly, combining either the R₁ or the R₂ GP with the SN degrades the result. Moreover, it leads to unstable training (see Figures 4(b) and (d)). This result implies that R₁ and R₂ GPs do not work well with the SN.
- Using the one-side GPs instead of their two-side counterparts increase the error rates by 0.1–9.5%. (We report the results for the one-side GPs in Appendix D due to page limit.)

¹This is possibly because the R₁ and the R₂ gradient penalties encourage D to have small gradients, and thus the gradients for both D and G might vanish when p_g and p_d are close enough.

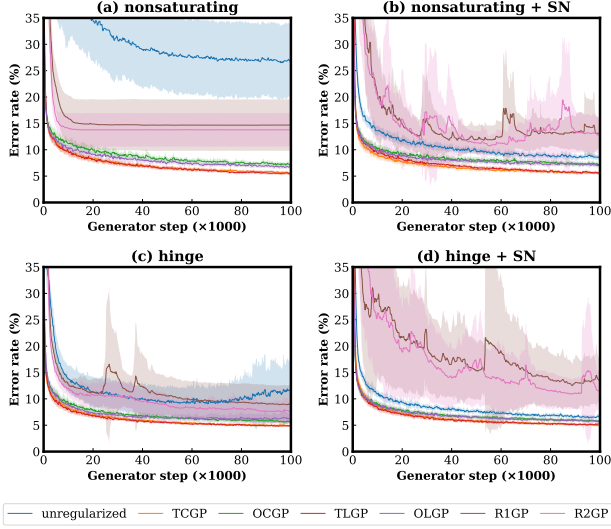


Figure 4. Error rates along the training progress for the nonsaturating and the hinge losses with common regularization approaches. The shaded regions represent the standard deviations over ten runs. The models are evaluated every 100 steps and the results are smoothed by a 5-point median filter. Best viewed in color.

We also note that some combinations result in remarkably high error rates, e.g., “least squares loss + R_1 GP”, “least squares loss + R_2 GP” and “classic minimax loss + R_1 GP”.

In sum, according to the overall performance and the robustness to different settings, for the component functions, *we recommend the hinge, the asymmetric and the two relativistic losses*. We note that these functions also feature lower computation costs as all their components functions are piecewise linear (see Table 1 and Section 3.6). For the regularization approaches, *we recommend the two-side coupled and the two-side local gradient penalties*.

We also conduct the imbalanced dataset test (see Section 4) on the two imbalanced datasets described in Section 5.1 to compare the regularization approaches. We use the classic nonsaturating loss. As shown in Table 5, the error rates increase as the level of imbalance increases. The two-side local GP achieve the lowest error rates across all three datasets. The error rates for the R_1 and the R_2 GPs increase significantly when the dataset goes imbalanced.

5.3.1. EFFECTS OF THE LIPSCHITZ CONSTANTS

In this experiment, we examine the effects of the Lipschitz constant (k) used in the coupled and the local GPs (see Table 2). We use the classic nonsaturating loss here. We report in Figure 5 the results for $k = 0.01, 0.1, 1, 10, 100$. We can see that the error rate increases as k goes away from 1.0, suggesting that $k = 1$ is indeed a good default value. Moreover, the two-side GPs are more sensitive to k than their one-side counterparts.

	standard	imbalanced	very imbalanced
TCGP	5.64 ± 0.23	7.09 ± 0.64	8.12 ± 0.31
OCGP	7.20 ± 0.39	8.86 ± 0.65	10.23 ± 0.75
TLGP	5.51 ± 0.27	6.94 ± 0.28	8.10 ± 0.55
OLGP	6.92 ± 0.21	8.63 ± 0.75	10.21 ± 0.52
R_1 GP	14.67 ± 4.86	18.66 ± 5.60	27.90 ± 9.59
R_2 GP	13.80 ± 3.20	15.70 ± 2.07	29.97 ± 12.4

Table 5. Error rates (%) for different gradient penalties (using the nonsaturating loss) on datasets with different levels of imbalance.

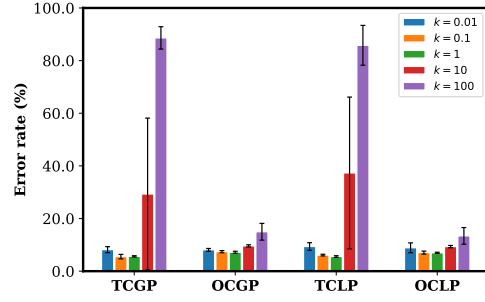


Figure 5. Effects of the Lipschitz constant k . Best viewed in color.

We note that [Petzka et al. \(2018\)](#) suggested that the one-side coupled GP are preferable to the two-side version and showed empirically that the former has more stable behaviors. However, we observe in our experiments that the two-side penalties usually lead to faster convergence to lower error rates compared to the one-side penalties.²

5.3.2. EFFECTS OF THE PENALTY WEIGHTS

We then examine the effects of the penalty weights (λ) for the R_1 and the R_2 GPs (see (4)). We consider the classic nonsaturating, the Wasserstein and the hinge losses. We present in Figure 6 the results for $\lambda = 0.01, 0.1, 1, 10, 100$. We can see that the R_1 GP tends to outperform the R_2 GP, while they are both sensitive to the value of λ . Hence, future research should run hyperparameter search for λ to find out its optimal value. When the spectral normalization is not used, the hinge loss is less sensitive to λ than the other two losses. However, when spectral normalization is used, the error rate increases as λ increases, which again implies that the R_1/R_2 GPs and the SN do not work well together.

5.4. On different generator loss functions

As discussed in Section 3.5, we also aim to examine the effects of the generator loss function $h(\cdot)$. We consider the classic and the hinge losses for the discriminator and the

²A possible reason is that as p_g move towards p_d , the gradients for G become smaller (and eventually zero when $p_d = p_g$), which can slow down the training. The two-side penalties can alleviate this by encouraging the norm of the gradients to be a fixed value.

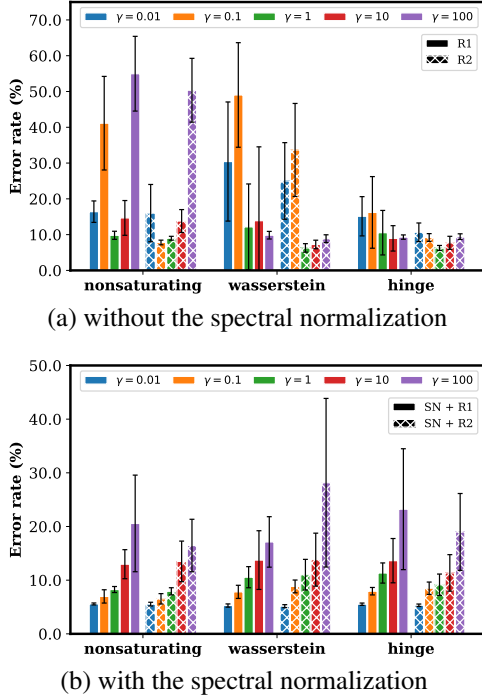


Figure 6. Effects of the penalty weight (λ). Best viewed in color.

following three generator loss functions: minimax (**M**)— $h(x) = g(x)$, nonsaturating (**N**)— $h(x) = \log(1 + e^{-x})$, and linear (**L**)— $h(x) = -x$. We report the results in the first six rows of Table 4. For the classic discriminator loss, we see no single winner among the three generator loss functions across all the regularization approaches, which implies that the heuristics behind these alternative losses might not be true. For the hinge discriminator loss, the minimax generator loss is robust to different regularization approaches and achieves three lowest and four lowest-three scores. Hence, we recommend to use hinge loss for the discriminator and minimax loss for the generator as the overall best choice according to our experimental results.

5.5. Effects of the momentum terms of the optimizers

We observe a trend towards using smaller momentum (Radford et al., 2016) or even no momentum (Arjovsky et al., 2017; Gulrajani et al., 2017; Miyato et al., 2018; Brock et al., 2018) in GAN training. Hence, we would also like to examine the effects of momentum terms in the optimizers with the proposed framework. As suggested by Gidel et al. (2018), we also include a negative momentum value of -0.5 . We use the classic nonsaturating loss and the SN along with the coupled GPs for regularization. Figure 7 shows the results for all combinations of $\beta_1 = -0.5, 0.0, 0.5, 0.9$ for G and D . We can see that for the two-side coupled GP, using larger momenta in both G and D leads to lower error rates, while there is no specific trend for the one-side coupled GP.

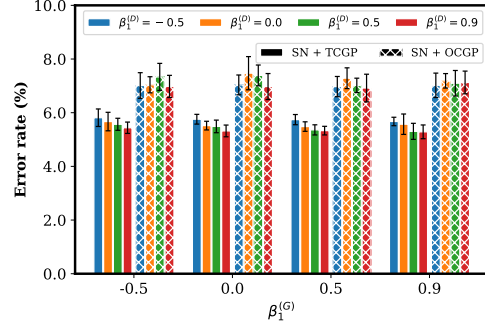


Figure 7. Effects of the momentum terms (β_1) in the optimizers.

6. Discussions and Conclusions

In this paper, we have shown in theory what certain types of component functions form a valid adversarial loss. We have also introduced a new framework called DANTest for comparing adversarial losses. With DANTest, we systematically compared combinations of different component functions and regularization approaches to decouple their effects. Our empirical results show that there is no single winning component functions or regularization approach across all different settings. Our theoretical and empirical results can together serve as a reference for choosing or designing adversarial training objectives in future research.

As compared to the commonly used metrics for evaluating generative models, such as the Inception Score (Salimans et al., 2016) and Frchet Inception Distance (Heusel et al., 2017) adopted in Lucic et al. (2018) and Kurach et al. (2018), the DANTest is simpler and is easier to control and extend. This allows us to easily evaluate new adversarial losses. However, we note that while the discriminator in a DAN is trained to optimize the same objectives as in a conditional GAN, the generators in the two models actually work in opposite ways ($\mathcal{X} \rightarrow \mathcal{Z}$ in a DAN versus $\mathcal{Z} \rightarrow \mathcal{X}$ in a GAN). Hence, it is unclear whether the empirical results can be generalized to conditional and unconditional GANs. Nonetheless, recent work has also adapted adversarial losses to plenty discriminative models (e.g., image-to-image translation (Isola et al., 2017) and image super-resolution (Ledig et al., 2017)). Therefore, it is worth investigating the behaviors of adversarial losses in different scenarios.

In addition, our theoretical analysis provides a new perspective on adversarial losses and reveals a large class of component functions valid for adversarial losses. We note that Nowozin et al. (2016) has also shown a certain class of component functions can result in theoretically valid adversarial losses. However, in their formulations, the component functions f and g are not independent of each other as they considered only the f -divergences. A future direction is to investigate the necessary and sufficient conditions for the existence and the uniqueness of a Nash equilibrium.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.
- Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 214–223, 2017.
- Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv:1607.06450*, 2016.
- Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv:1809.11096*, 2018.
- Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. Mode regularized generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2017.
- dos Santos, C. N., Wadhawan, K., and Zhou, B. Learning loss functions for semi-supervised learning via discriminative adversarial networks. *NeurIPS Workshop on Learning with Limited Labeled Data*, 2017.
- Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A. M., Mohamed, S., and Goodfellow, I. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In *International Conference on Learning Representations (ICLR)*, 2018.
- Gidel, G., Hemmat, R. A., Pezeshki, M., Lepriol, R., Huang, G., Lacoste-Julien, S., and Mitliagkas, I. Negative momentum for improved game dynamics. In *NeurIPS Workshop on Smooth Games Optimization and Machine Learning*, 2018.
- Goodfellow, I. NIPS 2016 tutorial: Generative adversarial networks. In *Thirtieth Conference on Neural Information Processing Systems (NeurIPS)*, 2016.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, 2014.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 5767–5777, 2017.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 6626–6637, 2017.
- Ioffe, S. and Szegedy, C. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.
- Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial nets. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- Jolicœur-Martineau, A. The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734*, 2018.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.
- Kodali, N., Abernethy, J., Hays, J., and Kira, Z. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215*, 2017.
- Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. The GAN landscape: Losses, architectures, regularization, and normalization. In *International Conference on Learning Representations (ICLR)*, 2018.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A. P., Tejani, A., Totz, J., Wang, Z., and Shi, W. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, 2017.
- Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 2203–2213, 2017.
- Lim, J. H. and Ye, J. C. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017.
- Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. Are GANs created equal? a large-scale study. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 698–707, 2018.

- Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for GANs do actually converge? In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3481–3490, 2018.
- Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014.
- Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.
- Mroueh, Y. and Sercu, T. Fisher GAN. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 2513–2523, 2017.
- Mroueh, Y., Sercu, T., and Goel, V. McGan: Mean and covariance feature matching GAN. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 2527–2535, 2017.
- Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pp. 271–279, 2016.
- Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of wasserstein gans. In *International Conference on Learning Representations (ICLR)*, 2018.
- Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pp. 2234–2242, 2016.
- Tran, D., Ranganath, R., and Blei, D. M. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 5523–5533, 2017.

Appendices

A. Proofs of the Theorems

Theorem 1. *If Property 1 holds, then for any $\gamma \in [0, 1]$, $\psi(\gamma) + \psi(1 - \gamma) \geq 2\psi(\frac{1}{2})$.*

Proof. Since Property 1 holds, we have for any fixed p_d ,

$$L_G \geq L_G \big|_{p_g=p_d}. \quad (1)$$

Let us consider

$$p_d(\mathbf{x}) = \gamma \delta(\mathbf{x} - \mathbf{s}) + (1 - \gamma) \delta(\mathbf{x} - \mathbf{t}), \quad (2)$$

$$p_g(\mathbf{x}) = (1 - \gamma) \delta(\mathbf{x} - \mathbf{s}) + \gamma \delta(\mathbf{x} - \mathbf{t}). \quad (3)$$

for some $\gamma \in [0, 1]$ and $\mathbf{s}, \mathbf{t} \in \mathcal{X}, \mathbf{s} \neq \mathbf{t}$. Then, we have

$$L_G \big|_{p_g=p_d} \quad (4)$$

$$= \max_D \int_{\mathbf{x}} p_d(\mathbf{x}) f(D(\mathbf{x})) + p_d(\mathbf{x}) g(D(\mathbf{x})) d\mathbf{x} \quad (5)$$

$$= \max_D \int_{\mathbf{x}} p_d(\mathbf{x}) (f(D(\mathbf{x})) + g(D(\mathbf{x}))) d\mathbf{x} \quad (6)$$

$$= \max_D \int_{\mathbf{x}} ((\gamma \delta(\mathbf{x} - \mathbf{s}) + (1 - \gamma) \delta(\mathbf{x} - \mathbf{t})) (f(D(\mathbf{x})) + g(D(\mathbf{x})))) d\mathbf{x} \quad (7)$$

$$= \max_D (\gamma(f(D(\mathbf{s})) + g(D(\mathbf{s}))) + (1 - \gamma)(f(D(\mathbf{t})) + g(D(\mathbf{t})))) \quad (8)$$

$$= \max_{y_1, y_2} (\gamma(f(y_1) + g(y_1)) + (1 - \gamma)(f(y_2) + g(y_2))) \quad (9)$$

$$= \max_{y_1} \gamma(f(y_1) + g(y_1)) + \max_{y_2} (1 - \gamma)(f(y_2) + g(y_2)) \quad (10)$$

$$= \max_y f(y) + g(y) \quad (11)$$

$$= 2\psi(\frac{1}{2}). \quad (12)$$

Moreover, we have

$$L_G = \max_D \int_{\mathbf{x}} p_d(\mathbf{x}) f(D(\mathbf{x})) + p_g(\mathbf{x}) g(D(\mathbf{x})) d\mathbf{x} \quad (13)$$

$$= \max_D \int_{\mathbf{x}} ((\gamma \delta(\mathbf{x} - \mathbf{s}) + (1 - \gamma) \delta(\mathbf{x} - \mathbf{t})) f(D(\mathbf{x})) + ((1 - \gamma) \delta(\mathbf{x} - \mathbf{s}) + \gamma \delta(\mathbf{x} - \mathbf{t})) g(D(\mathbf{x}))) d\mathbf{x} \quad (14)$$

$$= \max_D (\gamma f(D(\mathbf{s})) + (1 - \gamma) f(D(\mathbf{t})) + (1 - \gamma) g(D(\mathbf{s})) + \gamma g(D(\mathbf{t}))) \quad (15)$$

$$= \max_{y_1, y_2} (\gamma f(y_1) + (1 - \gamma) g(y_1)) + (1 - \gamma) f(y_2) + \gamma g(y_2) \quad (16)$$

$$= \max_{y_1} \gamma f(y_1) + (1 - \gamma) g(y_1) + \max_{y_2} (1 - \gamma) f(y_2) + \gamma g(y_2) \quad (17)$$

$$= \psi(\gamma) + \psi(1 - \gamma). \quad (18)$$

(Note that we can obtain (9) from (8) and (15) from (16) because D can be any function and thus $D(\mathbf{s})$ is independent of $D(\mathbf{t})$.)

As (1) holds for any fixed p_d , by substituting (12) and (18) into (1), we get

$$\psi(\gamma) + \psi(1 - \gamma) \geq 2\psi(\frac{1}{2}) \quad (19)$$

for any $\gamma \in [0, 1]$, which concludes the proof. \square

Theorem 2. *If Property 2 holds, then for any $\gamma \in [0, 1] \setminus \{\frac{1}{2}\}$, $\psi(\gamma) + \psi(1 - \gamma) > 2\psi(\frac{1}{2})$.*

Proof. Since Property 2 holds, we have for any fixed p_d ,

$$L_G \big|_{p_g \neq p_d} > L_G \big|_{p_g=p_d}. \quad (20)$$

Following the proof of Theorem 1, consider

$$p_d(\mathbf{x}) = \gamma \delta(\mathbf{x} - \mathbf{s}) + (1 - \gamma) \delta(\mathbf{x} - \mathbf{t}), \quad (21)$$

$$p_g(\mathbf{x}) = (1 - \gamma) \delta(\mathbf{x} - \mathbf{s}) + \gamma \delta(\mathbf{x} - \mathbf{t}), \quad (22)$$

for some $\gamma \in [0, 1]$ and some $\mathbf{s}, \mathbf{t} \in \mathcal{X}, \mathbf{s} \neq \mathbf{t}$. It can be easily shown that $p_g = p_d$ if and only if $\gamma = \frac{1}{2}$.

As (20) holds for any fixed p_d , by substituting (21) and (22) into (20), we get

$$\psi(\gamma) + \psi(1 - \gamma) > 2\psi(\frac{1}{2}), \quad (23)$$

for any $\gamma \in [0, 1] \setminus \{\frac{1}{2}\}$, concluding the proof. \square

Theorem 3. *If $\psi(\gamma)$ has a global minimum at $\gamma = \frac{1}{2}$, then Property 1 holds.*

Proof. First, we see that

$$L_G \big|_{p_g=p_d} \quad (24)$$

$$= \max_D \int_{\mathbf{x}} p_d(\mathbf{x}) f(D(\mathbf{x})) + p_d(\mathbf{x}) g(D(\mathbf{x})) d\mathbf{x} \quad (25)$$

$$= \max_y \int_{\mathbf{x}} p_d(\mathbf{x}) f(y) + p_d(\mathbf{x}) g(y) d\mathbf{x} \quad (26)$$

$$= \max_y \int_{\mathbf{x}} p_d(\mathbf{x}) (f(y) + g(y)) d\mathbf{x} \quad (27)$$

$$= \max_y (f(y) + g(y)) \int_{\mathbf{x}} p_d(\mathbf{x}) d\mathbf{x} \quad (28)$$

$$= \max_y f(y) + g(y) \quad (29)$$

$$= 2\psi(\frac{1}{2}). \quad (30)$$

On the other had, we have

$$L_G = \max_D \int_{\mathbf{x}} p_d(\mathbf{x}) f(D(\mathbf{x})) + p_g(\mathbf{x}) g(D(\mathbf{x})) d\mathbf{x} \quad (31)$$

$$= \max_y \int_{\mathbf{x}} p_d(\mathbf{x}) f(y) + p_g(\mathbf{x}) g(y) d\mathbf{x} \quad (32)$$

$$= \max_y \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \left(\frac{p_d(\mathbf{x}) f(y)}{p_d(\mathbf{x}) + p_g(\mathbf{x})} + \frac{p_g(\mathbf{x}) g(y)}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \right) d\mathbf{x} \quad (33)$$

$$= \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \max_y \left(\frac{p_d(\mathbf{x}) f(y)}{p_d(\mathbf{x}) + p_g(\mathbf{x})} + \frac{p_g(\mathbf{x}) g(y)}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \right) d\mathbf{x}. \quad (34)$$

Since $\frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \in [0, 1]$, we have

$$L_G = \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \psi \left(\frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \right) d\mathbf{x}. \quad (35)$$

As $\psi(\gamma)$ has a global minimum at $\gamma = \frac{1}{2}$, now we have

$$L_G \geq \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \psi(\frac{1}{2}) d\mathbf{x} \quad (36)$$

$$= \psi(\frac{1}{2}) \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) d\mathbf{x} \quad (37)$$

$$= 2\psi(\frac{1}{2}). \quad (38)$$

Finally, combining (30) and (38) yields

$$L_G \geq L_G \big|_{p_g=p_d}, \quad (39)$$

which holds for any p_d , thus concluding the proof. \square

Theorem 4. If $\psi(\gamma)$ has a unique global minimum at $\gamma = \frac{1}{2}$, then Property 2 holds.

Proof. Since $\psi(\gamma)$ has a unique global minimum at $\gamma = \frac{1}{2}$, we have for any $\gamma \in [0, 1] \setminus \frac{1}{2}$,

$$\psi(\gamma) > \psi(\frac{1}{2}). \quad (40)$$

When $p_g \neq p_d$, there must be some $\mathbf{x}_0 \in \mathcal{X}$ such that $p_g(\mathbf{x}_0) \neq p_d(\mathbf{x}_0)$. Thus, $\frac{p_d(\mathbf{x}_0)}{p_d(\mathbf{x}_0) + p_g(\mathbf{x}_0)} \neq \frac{1}{2}$, and thereby $\psi \left(\frac{p_d(\mathbf{x}_0)}{p_d(\mathbf{x}_0) + p_g(\mathbf{x}_0)} \right) > \psi(\frac{1}{2})$. Now, by (35) we have

$$L_G \big|_{p_g \neq p_d} \quad (41)$$

$$= \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \psi \left(\frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \right) d\mathbf{x} \quad (42)$$

$$> \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) \psi(\frac{1}{2}) d\mathbf{x} \quad (43)$$

$$= \psi(\frac{1}{2}) \int_{\mathbf{x}} (p_d(\mathbf{x}) + p_g(\mathbf{x})) d\mathbf{x} \quad (44)$$

$$= 2\psi(\frac{1}{2}). \quad (45)$$

Finally, combining (30) and (45) yields

$$L_G \big|_{p_g \neq p_d} > L_G \big|_{p_g=p_d}, \quad (46)$$

which holds for any p_d , thus concluding the proof. \square

Theorem 5. If $f'' + g'' \leq 0$ and there exists some y^* such that $f(y^*) = g(y^*)$ and $f'(y^*) = -g'(y^*) \neq 0$, then $\psi(\gamma)$ has a unique global minimum at $\gamma = \frac{1}{2}$.

Proof. First, we have by definition

$$\Psi(\gamma, y) = \gamma f(y) + (1 - \gamma) g(y). \quad (47)$$

By taking the partial derivatives, we get

$$\frac{\partial \Psi}{\partial \gamma} = f(y) - g(y), \quad (48)$$

$$\frac{\partial \Psi}{\partial y} = \gamma f'(y) + (1 - \gamma) g'(y), \quad (49)$$

$$\frac{\partial^2 \Psi}{\partial y^2} = \gamma f''(y) + (1 - \gamma) g''(y). \quad (50)$$

We know that there exists some y^* such that

$$f(y^*) = g(y^*), \quad (51)$$

$$f'(y^*) = -g'(y^*) \neq 0. \quad (52)$$

(i) By (48) and (49), we see that

$$\frac{\partial \Psi}{\partial \gamma} \big|_{y=y^*} = 0, \quad (53)$$

$$\frac{\partial \Psi}{\partial y} \big|_{(\gamma, y)=(\frac{1}{2}, y^*)} = 0. \quad (54)$$

Now, by (53) we know that Ψ is constant when $y = y^*$. That is, for any $\gamma \in [0, 1]$,

$$\Psi(\gamma, y^*) = \Psi(\frac{1}{2}, y^*). \quad (55)$$

(ii) Because $f'' + g'' \leq 0$, by (50) we have

$$\frac{\partial^2 \Psi}{\partial y^2} \big|_{\gamma=\frac{1}{2}} = \frac{1}{2} f''(y) + \frac{1}{2} g''(y) \quad (56)$$

$$\leq 0. \quad (57)$$

By (54) and (56), we see that y^* is a global minimum point of $\Psi|_{\gamma=\frac{1}{2}}$. Thus, we now have

$$\Psi(\tfrac{1}{2}, y^*) = \max_y \Psi(\tfrac{1}{2}, y) \quad (58)$$

$$= \psi(\tfrac{1}{2}). \quad (59)$$

(iii) By (49), we see that

$$\frac{\partial \Psi}{\partial y} \Big|_{y=y^*} = \gamma f'(y^*) + (1 - \gamma) g'(y^*) \quad (60)$$

$$= \gamma f'(y^*) + (1 - \gamma) (-f'(y^*)) \quad (61)$$

$$= (2\gamma - 1) f'(y^*). \quad (62)$$

Since $f'(y^*) \neq 0$, we have

$$\frac{\partial \Psi}{\partial y} \Big|_{y=y^*} \neq 0 \quad \forall \gamma \in [0, 1] \setminus \tfrac{1}{2}. \quad (63)$$

This shows that for any $\gamma \in [0, 1] \setminus \tfrac{1}{2}$, there must exists some y° such that

$$\Psi(\gamma, y^\circ) > \Psi(\gamma, y^*). \quad (64)$$

And by definition we have

$$\Psi(\gamma, y^\circ) < \max_y \Psi(\gamma, y) \quad (65)$$

$$= \psi(\gamma). \quad (66)$$

Hence, by (64) and (65) we get

$$\psi(\gamma) > \Psi(\gamma, y^*). \quad (67)$$

Finally, combining (54), (59) and (67) yields

$$\psi(\gamma) > \psi(\tfrac{1}{2}) \quad \forall \gamma \in [0, 1] \setminus \tfrac{1}{2}, \quad (68)$$

which concludes the proof. \square

B. More Graphs of the Ψ and ψ Functions

We show in Figure 1 the graphs of the ψ functions for different adversarial losses. Note that for the Wasserstein loss, the ψ function is only defined at $\gamma = 0.5$, where it takes the value of zero, and for the asymmetric loss, the ψ function is only defined when $\gamma > 0.5$, where it takes the value of zero. Hence, we do not include them in Figure 1.

We also present in Figure 2 the graphs of the Ψ functions for the ϵ -weighted versions of the classic, the Wasserstein and the hinge losses. Moreover, Figures 1(b) and (c) show the graphs of the ψ functions for the ϵ -weighted versions of the classic and the hinge losses, respectively.

C. Network Architectures

We present in Table 1 the network architectures for the generator and the discriminator used for all the experiments.

D. More Results

We report in Table 2 the results for the one-side coupled and local gradient penalties.

We also present in Figure 3 the results for the experiment on the momentum terms using the hinge loss.

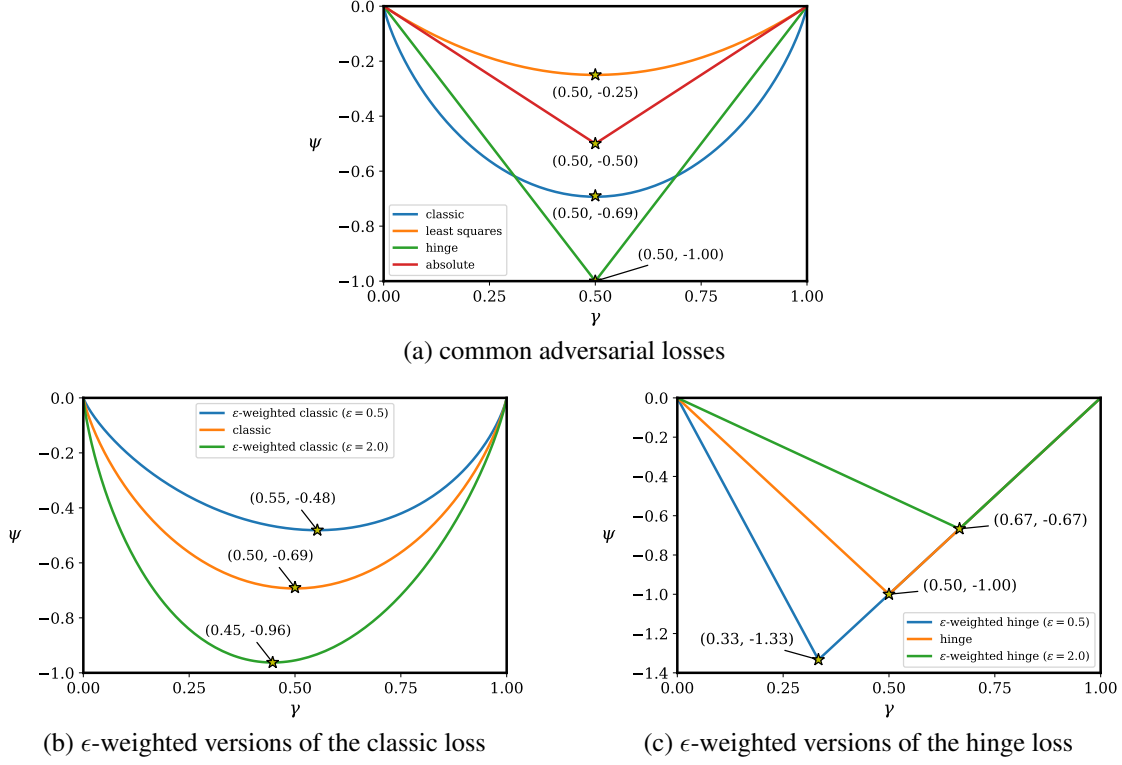


Figure 1. Graphs of the ψ functions for different adversarial losses. The star marks indicate their minima. Best viewed in color.

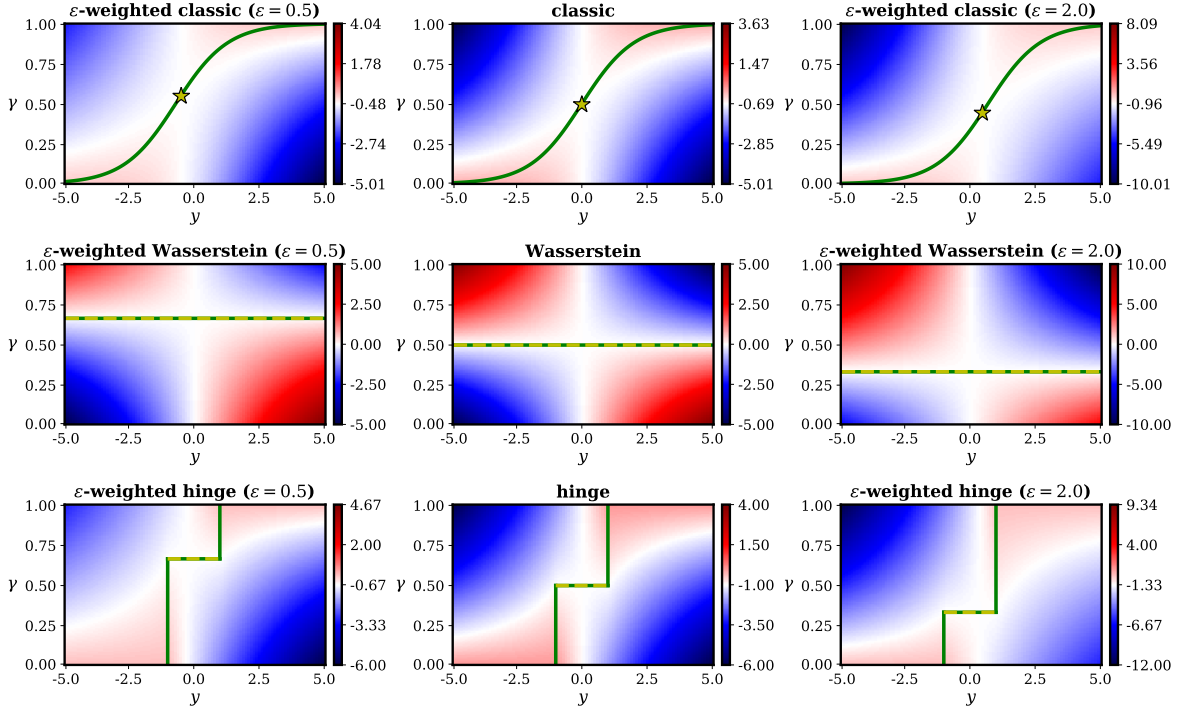


Figure 2. Graphs of the Ψ functions of different adversarial losses. The green lines show the domains of the ψ functions (i.e., the value(s) that y can take for different γ in the ψ function). The star marks, and any points on the yellow dashed lines, are the minimum points of ψ . The midpoints of the color maps are intentionally set to the minima of ψ (i.e., the values taken at the star marks or the yellow segments). Note that $\gamma \in [0, 1]$ and $y \in \mathbb{R}$, so we plot different portions of y where the characteristics of Ψ can be clearly seen.

Generator (G)				Discriminator (D)			
<i>conv</i>	32	3×3	3×3	<i>conv</i>	32	3×3	3×3
<i>conv</i>	64	3×3	3×3	<i>conv</i>	64	3×3	3×3
<i>maxpool</i>	-	2×2	2×2	<i>maxpool</i>	-	2×2	2×2
<i>dense</i>	128			<i>dense</i>	128		
<i>dense</i>	10			<i>dense</i>	1		

Table 1. Network architectures for the generator and the discriminator used for all the experiments. For the convolutional layer (*conv*), the values represent (from left to right): the number of filters, the kernel sizes and the strides. For the max pooling (*maxpool*) layer, the values represent (from left to right): the pool sizes and the strides. For the dense (*dense*) layer, the value indicates the number of nodes. The activation functions are ReLUs except for the last layer of the generator, which uses the softmax functions, and the last layer of the discriminator, which has no activation function.

	OCGP	OLGP	SN + OCGP	SN + OLGP
classic (M) (Goodfellow et al., 2014)	7.15 ± 0.77	6.95 ± 0.51	7.16 ± 0.31	6.86 ± 0.29
classic (N) (Goodfellow et al., 2014)	7.20 ± 0.39	6.98 ± 0.22	7.47 ± 0.62	7.15 ± 0.36
classic (L)	7.12 ± 0.61	7.00 ± 1.00	7.29 ± 0.35	7.18 ± 0.54
hinge (M)	5.82 ± 0.31	7.33 ± 1.35	5.80 ± 0.24	5.83 ± 0.20
hinge (N)	5.69 ± 0.30	7.88 ± 1.33	5.92 ± 0.36	5.74 ± 0.27
hinge (L) (Lim & Ye, 2017; Tran et al., 2017)	5.77 ± 0.29	6.22 ± 1.04	5.77 ± 0.30	5.82 ± 0.20
Wasserstein (Arjovsky et al., 2017)	7.60 ± 3.02	13.34 ± 1.49	6.35 ± 0.43	6.06 ± 0.45
least squares (Mao et al., 2017)	7.99 ± 0.35	8.06 ± 0.49	8.43 ± 0.50	8.31 ± 0.52
relativistic (Jolicoeur-Martineau, 2018)	8.03 ± 3.32	9.41 ± 2.90	6.18 ± 0.29	6.03 ± 0.24
relativistic hinge (Jolicoeur-Martineau, 2018)	10.70 ± 2.51	14.17 ± 1.79	5.42 ± 0.33	5.42 ± 0.33
absolute	5.95 ± 0.19	5.88 ± 0.41	6.22 ± 0.25	6.08 ± 0.32
asymmetric	5.85 ± 0.35	7.57 ± 0.98	6.21 ± 0.34	5.92 ± 0.37

Table 2. Error rates (%) for different adversarial losses and regularization approaches, on the standard dataset. OCGP and OLCP stand for the one-side coupled and the one-side local gradient penalties, respectively. SN stands for the spectral normalization. M, N and L stand for the minimax, the nonsaturating and the linear loss functions used in the generator, respectively. Underlined and bold fonts indicate respectively entries with the lowest and lowest-three mean error rates per column.

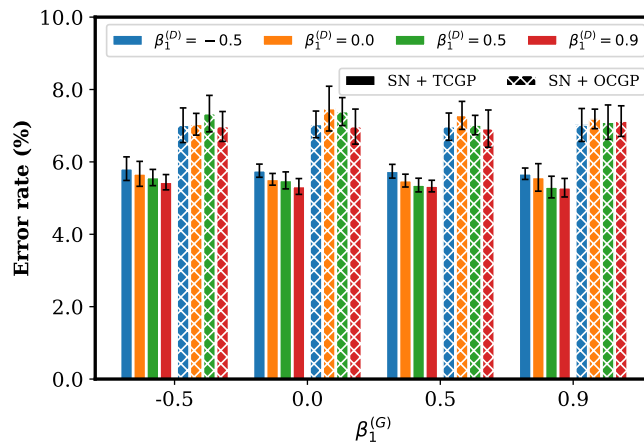


Figure 3. Effects of the momentum terms (β_1) in the optimizers for the hinge loss. Best viewed in color.