Received: 22 January 2019 | Accepted: 1 February 2019
DOI: 10.1002/sta4.224

WILEY

ORIGINAL ARTICLE

# Calibrated adversarial algorithms for generative modelling

Zhiqiang Tan[1] | Yunfu Song[2] | Zhijian Ou[2]

[1]Department of Statistics, Rutgers University, Piscataway, New Jersey
[2]Department of Electronic Engineering, Tsinghua University, Beijing, China

**Correspondence**
Zhiqiang Tan, Department of Statistics, Rutgers University, Piscataway, NJ 08854.
Email: ztan@stat.rutgers.edu

Zhijian Ou, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
Email: ozj@tsinghua.edu.cn

Generative adversarial networks are useful for unsupervised learning but may be difficult to train. We study a class of adversarial algorithms based on *f*-divergence minimization and provide an extension by allowing two objective functions instead of one to be chosen (hence calibrated) for updating the discriminator and the generator, respectively. The extension is derived and justified from theoretical analysis, which identifies specific objective functions for achieving stable gradients in the corresponding updates. Our experiments on synthetic data and MNIST and CIFAR-10 datasets demonstrate that the proposed method consistently achieves competitive or superior results when compared with various existing methods.

**KEYWORDS**

generative adversarial network, *f*-divergence, Kullback–Liebler divergence, minimum divergence estimation

## 1 | INTRODUCTION

Generative adversarial networks (GANs) are a useful class of methods for unsupervised learning, as proposed in Goodfellow et al. (2014) and later improved and extended in various directions, for example, Nowozin, Cseke, and Tomioka (2016), Salimans et al. (2016), and others. A general idea of GANs is to employ two neural networks (or other statistical models): a generator $p_\theta(x)$ that produces a sample $x$ by transforming a noise vector $z$ as $x = g_\theta(z)$ and a discriminator $D_\beta(x)$ that assigns class probabilities to distinguish the generated samples ("fake data") from real data. The two networks are trained against each other, until a certain equilibrium is reached.

Training of GANs, however, remains delicate. We study a class of adversarial algorithms based on *f*GAN (Nowozin et al., 2016) and provide an extension, called calibrated GAN (Cal-GAN), by allowing two objective functions instead of one to be chosen (hence calibrated) for updating the discriminator and the generator, respectively. Cal-GAN can be employed, similarly as *f*GAN, to approximately minimize any *f*-divergence (Ali & Silvey, 1966), with a suitable objective used for the generator. Examples include the Kullback–Liebler (KL) divergence, $\mathrm{KL}(p_* \| p_\theta)$, and the reverse KL divergence, defined as $\mathrm{KL}(p_\theta \| p_*)$, where $p_*(x)$ denotes the data distribution (see Table S1 in the Supporting Information).

Although various algorithms have been proposed using two objectives for GANs (e.g., Goodfellow et al., 2014; Lim & Ye, 2017; Poole, Alemi, Sohl-Dickstein, & Angelova, 2016; Zhao, Mathieu, & LeCun, 2017), our *main contribution* in developing Cal-GAN is to (a) provide theoretical analysis on stability of gradients depending on the training objectives for the discriminator and the generator and (b) identify specific objective functions to achieve stable gradients. In addition, our analysis also discovers new connections and inaccurate claims in the literature (see Section 2).

The resulting Cal-GAN algorithm involves using maximum likelihood for training the discriminator and the reverse KL or mixed KL divergence for training the generator. For the reverse KL divergence, the population version of our method reduces to

$$\begin{cases} \max_\beta \; K_{\mathrm{GAN}}(\theta, \beta) & \text{with } \theta \text{ fixed}, \quad (1a) \\ \max_\theta \; -E_{p_\theta(x)}\{h_\beta(x)\} & \text{with } \beta \text{ fixed}, \quad (1b) \end{cases}$$

where $K_{\mathrm{GAN}}(\theta, \beta)$ is the original GAN objective, defined in (2) later, and $h_\beta(x) = \log\{D_\beta(x)/(1 - D_\beta(x))\}$ with $D_\beta(x)$ representing the probability that a sample $x$ comes from real data. Our method can also be used with a mixed KL divergence, as discussed in Section 4.4.

There are interesting differences known between minimizing *f*-divergences (Huszár, 2015; Minka, 2005; Theis, Den Oord, & Bethge, 2016). In particular, minimizing KL (i.e., maximum likelihood) tends to cover all modes of the data, whereas minimizing the reverse KL often leads to concentration around a few modes, which seems helpful for generating sharp, realistic samples. But these differences may not fully explain the

behaviour of GANs (Goodfellow, 2017), as also seen from our experiments. Our use of the reverse or mixed KL divergence is currently mainly motivated by the gradient stability obtained.

We present numerical experiments with Gaussian mixture data and two image datasets: MNIST and CIFAR-10. Our method is found to achieve competitive or superior results when compared with various existing methods, with or without using additional training techniques.

In practice, training of GANs involves a number of considerations, including network architectures, objective functions, and normalization and regularization schemes. Our work on the choices of objective functions is compatible and can be synthesized with other techniques, for example, spectral normalization (Miyato, Kataoka, Koyama, & Yoshida, 2018).

## 2 | RELATED WORK

GANs have been extensively studied. For space limitation, we discuss directly related work to ours.

### 2.1 | Relation to *f*GAN

Our method presents a further development of *f*GAN (Nowozin et al., 2016), which involves a single objective function and provides no analysis of gradient stability. Our reformulation of *f*GAN (Algorithm 1) is closely related to Uehara, Sato, Suzuki, Nakayama, and Matsuo (2016) and Mohamed and Lakshminarayanan (2017) from the perspective of density ratio estimation. This algorithm was suggested as one of the three options in Uehara et al. (2016). But we show in Appendix B.1 that, in general, *none* of the other two (preferred) options in Uehara et al. (2016) and another option in Mohamed and Lakshminarayanan (2017) leads to minimization of the target *f*-divergence, even with a nonparametric discriminator.

### 2.2 | Objectives and gradient stability

Although the issue of gradient instability has been widely recognized and various objective functions have been proposed (e.g., Goodfellow et al., 2014; Poole et al., 2016), there seems to be limited work on theoretical analysis except the paper by Arjovsky and Bottou (2017). Our analysis not only yields Theorem 2.4 in Arjovsky and Bottou (2017) on vanishing gradients in GAN as a special case but also leads to *a sharper result* than Theorem 2.5 in Arjovsky and Bottou (2017) and *a different conclusion* than Theorem 2.6 in Arjovsky and Bottou (2017) on the behaviour of GAN with log*D* trick. See Appendix B.3 for details.

### 2.3 | Divergences and disjoint manifolds

Our use of the reverse or mixed KL divergence also seems to be at odds with the criticism in Arjovsky and Bottou (2017) and Arjovsky, Chintala, and Bottou (2017) that minimization of popular divergences such as KL and reverse KL divergences would lead to unstable training of the generator, because these divergences are maxed out when the data and generator's distributions are disjoint. But this analysis assumes that the discriminator is fully trained to optimality for any fixed generator, which is practically violated when the discriminator and generator are alternately updated by 1 or a few steps of stochastic gradient descent. In Figure 1, we borrow a simple example in Arjovsky et al. (2017) to illustrate appropriate behaviour of our method even when the generator's and data distributions live on disjoint manifolds.

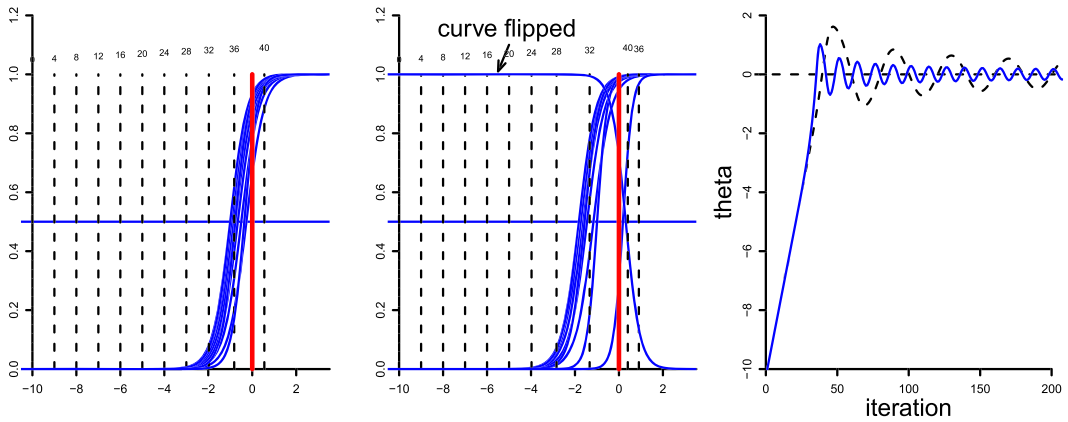



**FIGURE 1** Learning parallel lines in Example 1 of Arjovsky et al. (2017), using program (1), Cal-GAN with reserve KL. Left: the learned lines (dashed) moving from −10 to the data at 0 (red solid line) and the discrimination probabilities (blue solid lines) until the 40th iteration, where the discriminator and generator are each updated by 1 step of gradient descent, with learning rates 1 and 0.1, respectively. The *y*-axis is used for both dashed and blue lines. The number above each learned line is the corresponding number of iterations. Middle: same as Left, except that the discriminator is updated by 10 steps of gradient descent for each step of updating the generator. Right: the trajectories of $\theta$ over 200 iterations (dashed or solid: 1 or 10 discriminator updates for each generator update)

### 2.4 | Reverse and mixed KL

Program (1) was previously derived in Huszár (2016), by an ad hoc modification of the log$D$ trick in GAN (Goodfellow et al., 2014) to obtain reverse KL as an objective function. Our work handles any $f$-divergence and provides supportive analysis for use of the reverse or mixed KL divergence. Program (1) is also equivalent to a method in Chen et al. (2018), but which was *incorrectly* claimed to result in minimization of the symmetric KL divergence (see Section 4.4). The method in Nguyen, Le, Vu, and Phung (2017) can be treated as an extension of $f$GAN with mixed KL to use two discriminators (see Appendix A).

### 2.5 | Hinge and energy-based GAN

Hinge GAN is similar to program (1), but with $K_{\text{GAN}}(\theta, \beta)$ in (1a) replaced by the hinge loss (Lim & Ye, 2017; Miyato et al., 2018). The derivation in Lim and Ye (2017) is heuristic, based on geometry of the hinge loss. We show in Appendix B.4 that this method leads to minimization of the total-variation distance for the generator *not* the reverse KL as stated in Miyato et al. (2018). This implies that hinge GAN is theoretically equivalent to energy-based GAN (Zhao et al., 2017), which was previously shown in Arjovsky et al. (2017) to optimize the total-variation distance.

## 3 | ILLUSTRATION

Consider learning parallel lines as in Example 1 of Arjovsky et al. (2017), where the data distribution is uniform on $\{(0, x_2) : 0 \le x_2 \le 1\}$ and the generator is defined by $g_\theta(z) = (\theta, z)$ with $z \sim \text{Unif}(0, 1)$, for a parameter $\theta$, initialized at $-10$. A discriminator is defined by the logistic regression: $h_\beta(x) = \beta_0 + \beta_1 x_1$, with $(\beta_0, \beta_1)$ initialized at $(0, 0)$. Although the data and generator's distributions are disjoint, Figure 1 (right) shows that direct implementation of program (1) by gradient descent yields a sequence of estimates of $\theta$ converging properly to 0.

The left and middle plots in Figure 1 show how the discriminator is changed during training. Each blue curve gives predicted probabilities from the discriminator and may increase from 0 to 1 in one direction of $x$ or the other, by the simple logistic discriminator defined. For the middle plot, when the learned line is to the right of $x = 0$ at the 40th iteration, the blue curve reverses its increasing direction ("curve flipped") as expected. For the left plot, the curve is not flipped, even when the learned line is to the right of $x = 0$ at 40th iteration. The difference is caused by the fact that the discriminator is updated by 10 steps of gradient descent in the middle plot, but by one step of gradient descent in the left plot. Such desynchronization between the discriminator and the generator also explains the oscillation in the trajectories of $\theta$ in the right plot.

See Section IV of the Supporting Information for the corresponding plots from WGAN (Arjovsky et al., 2017) and hinge GAN (Lim & Ye, 2017) and further discussion about oscillation in the estimates of $\theta$ from those methods.

## 4 | THEORY AND METHOD

Let $p_*(x)$ be a data distribution (or density) and $p_\theta(x)$ be a model density with parameters $\theta$. For generative modelling, assume that $p_\theta(x)$ is defined through a (differentiable) transformation, $x = g_\theta(z)$, from a noise vector $z \sim p(z)$, with a known prior density $p(z)$. In various applications, $z$ is often a vector of independent, standard normal variables, and $g_\theta(z)$ is a feedforward neural network.

### 4.1 | Background: GAN and $f$GAN

Recently, adversarial learning has emerged as a useful approach for generative modelling as in GAN (Goodfellow et al., 2014). The population version of GAN training is proposed as a two-player game:

$$\min_\theta \max_\beta \; K_{\text{GAN}}(\theta, \beta) = E_{p_*(x)}\left[\log\{D_\beta(x)\}\right] + E_{p(z)}\left[\log\{1 - D_\beta(g_\theta(z))\}\right], \tag{2}$$

where $D_\beta(x)$ is a discriminator depending on parameters $\beta$, representing the probability that an observation $x$ comes from $p_*(x)$ rather than $p_\theta(x)$. Formally, let $y \in \{0, 1\}$ be a Bernoulli $(1/2)$ variable and $x \sim p_*(x)$ or $x \sim p_\theta(x)$ if $y = 1$ or 0. Consider a logistic discrimination model,

$$P(y = 1|x) = D_\beta(x) = \sigma(h_\beta(x)), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function and $h_\beta(x)$ is a neural network without range restriction. If $D_\beta(x)$ is sufficiently rich and well trained, then GAN minimizes the Jensen–Shannon (JS) divergence for learning $p_\theta$.

In practice, however, a modification of GAN is often used with a log$D$ trick, to overcome vanishing gradients in $\theta$ when the discriminator is confident, for example, at the early stage of training (Goodfellow et al., 2014). This method, denoted as GAN2 here, amounts to solving the following problem:

$$\begin{cases} \max_\beta K_{\text{GAN}}(\theta, \beta) \text{ with } \theta \text{ fixed}, \\ \min_\theta -E_{p(z)}\left[\log\{D_\beta(g_\theta(z))\}\right] \text{ with } \beta \text{ fixed}. \end{cases} \tag{4}$$

Although the log$D$ trick helps to alleviate vanishing gradients, the criterion that is minimized by GAN2 for learning $p_\theta$ is no longer the JS divergence (Arjovsky & Bottou, 2017), as further discussed in Appendix B.3.

From the perspective of variational divergence estimation (Nguyen, Wainwright, & Jordan, 2010), Nowozin et al. (2016) proposed a broad class of adversarial algorithms, called $f$GAN, which include GAN and, with a similar trick, GAN2. For a convex function $f : \mathbb{R}_+ \to \mathbb{R}$, the $f$-divergence between $p_*$ and $p_\theta$ is (Ali & Silvey, 1966)

$$\mathcal{D}_f(p_*||p_\theta) = \int p_\theta(x) f\left(\frac{p_*(x)}{p_\theta(x)}\right) \mathrm{d}x. \tag{5}$$

Let $f^*$ be the Fenchel conjugate of $f$, that is, $f^*(t) = \sup_{u\in\mathbb{R}_+}\{ut - f(u)\}$. The population version of $f$GAN training solves the following saddle-point problem:

$$\min_\theta \max_\beta \; K_{f\mathrm{GAN}}(\theta,\beta) = E_{p_*(x)}\{T_\beta(x)\} - E_{p_\theta(x)}\{f^*(T_\beta(x))\}, \tag{6}$$

where $T_\beta(x)$ is a variational function, taking values in the domain of $f^*$. By this restriction, $T_\beta(x)$ is represented as $T_\beta(x) = \tau_f(V_\beta(x))$, where $V_\beta(x)$ can be a neural network without range restriction on the output and $\tau_f : \mathbb{R} \to \mathrm{dom}_{f^*}$ is an activation function applied to $V_\beta(x)$. Nowozin et al. (2016) recommended individual choices of the activation function $\tau_f$ on a case-by-case basis for common divergences. Given enough capacity and training of $T_\beta(x)$, $f$GAN then effectively minimizes $\mathcal{D}_f(p_*||p_\theta)$ for learning $\theta$.

## 4.2 | Logit $f$GAN

We reformulate $f$GAN learning with a concrete choice of the variational function $T_\beta(x)$ depending on the logit $h_\beta(x)$ in discrimination model (3), hence called logit $f$GAN. This formulation puts $f$GAN back in the form of training a generator and a discriminator against each other as originally in GAN (Goodfellow et al., 2014). Moreover, this formulation facilitates our gradient analysis in Section 4.3. In general, the gradients for the generator and the discriminator in $f$GAN would depend on both the choice of the variational function and the choice of $f$-divergence.

For a convex function $f$ as in (5), consider the objective function

$$K_f(\theta,\beta) = E_{p_*(x)}\{f'(U_\beta(x))\} - E_{p_\theta(x)}\{U_\beta(x)f'(U_\beta(x)) - f(U_\beta(x))\}, \tag{7}$$

where $f'$ is the derivative of $f$, and $U_\beta(x) = \mathrm{e}^{h_\beta(x)}$, representing the odds $P(y = 1|x)/P(y = 0|x)$ under model (3) or equivalently, by the Bayes rule, the density ratio $p_*(x)/p_\theta(x)$. The objective $K_f(\theta,\beta)$ can be obtained from the $f$GAN objective in (6) by taking $T_\beta(x) = f'(\mathrm{e}^{h_\beta(x)})$ with $V_\beta(x) = h_\beta(x)$ and $\tau_f(v) = f'(\mathrm{e}^v)$ and then applying the relationship (Boyd & Vandenberghe, 2004, Section 3.3.2):

$$f^*(f'(u)) = uf'(u) - f(u), \quad u \in \mathbb{R}_+. \tag{8}$$

In other words, the objective (7) corresponds to the $f$GAN objective in (6) with a specific choice of the variational function $T_\beta(x)$, depending on discrimination model (3).

We stress that the objective $K_f(\theta,\beta)$ provides a proper objective function in training the discriminator (3) for any fixed $\theta$. In fact, by invoking the joint distribution of $(y, x)$, the objective $K_f(\theta,\beta)$ can be written as $K_f(\theta,\beta) = 2E\{L_f(y, D_\beta)\}$, where $L_f(y, D_\beta)$ is a discrimination (negative) loss:

$$L_f(y, D_\beta) = yf'(U_\beta(x)) - (1-y)\{U_\beta(x)f'(U_\beta(x)) - f(U_\beta(x))\}. \tag{9}$$

In particular, if $f(u) = u\log u - (u+1)\log(u+1)$, then (9) gives the log-likelihood in model (3), $y\log D_\beta(x) + (1-y)\log(1 - D_\beta(x))$, and (7) reduces to the GAN objective in (2). See Section II in the Supporting Information for further discussion on $f$-divergences and discrimination losses.

The population version of logit $f$GAN is defined by solving

$$\min_\theta \max_\beta \; K_f(\theta,\beta). \tag{10}$$

Algorithm 1 presents a procedure for solving (10) similarly as in Nowozin et al. (2016). If the discriminator is sufficiently rich and well trained, then logit $f$GAN approximately minimizes $\mathcal{D}_f(p_*||p_\theta)$ for learning $\theta$. See Appendix B.1 for a discussion about related algorithms in Uehara et al. (2016) and Mohamed and Lakshminarayanan (2017).

**Algorithm 1** Logit $f$GAN

**repeat**

**Sampling:** Draw a minibatch $\{x_1, \ldots, x_m\}$ from the data distribution $p_*(x)$ and a minibatch $\{z_1, \ldots, z_m\}$ from the noise prior $p(z)$.

**Updating:** Denote $\xi_i = g_\theta(z_i)$ and

$$\hat{K}_f(\theta,\beta) = \frac{1}{m}\sum_{i=1}^m f'(\mathrm{e}^{h_\beta(x_i)}) - \frac{1}{m}\sum_{i=1}^m \left\{\mathrm{e}^{h_\beta(\xi_i)}f'(\mathrm{e}^{h_\beta(\xi_i)}) - f(\mathrm{e}^{h_\beta(\xi_i)})\right\};$$

Update $\beta$ by ascending the gradient $\nabla_\beta \hat{K}_f(\theta,\beta)$; Update $\theta$ by descending the gradient $\nabla_\theta \hat{K}_f(\theta,\beta)$.

**until** convergence

## 4.3 | ACHIEVING STABLE GRADIENTS

With the explicit form of the objective function $K_f(\theta, \beta)$ in (7), we study how to choose $f$-divergences and discrimination losses to achieve stable gradients for the generator and the discriminator, respectively.

### 4.3.1 | Generator gradient

We examine the gradient of $K_f(\theta, \beta)$ with respect to $\theta$ for fixed $\beta$, which by using the chain rule can be shown to be

$$\nabla_\theta K_f(\theta, \beta) = -E_{p(z)}\left\{ f''(e^{h_\beta(x)})e^{2h_\beta(x)}\Big|_{x=g_\theta(z)} \times \nabla_\theta h_\beta(g_\theta(z)) \right\}, \tag{11}$$

under the exchange of differentiation and expectation, where $f''$ denotes the second-order derivative of $f$. The issue of vanishing gradients for GAN can be understood as follows, related to Theorem 2.4 in Arjovsky and Bottou (2017). If the discriminator is confident in well separating fake from real data (either because the generator is poor at the early stage of training or because the discriminator is trained "too much"), then the probability $D_\beta(x)$ tends to be close to 0, or equivalently, the log-odds $h_\beta(x)$ tend to $-\infty$ for fake data $x = g_\theta(z)$. For $f(u) = u\log u - (u+1)\log(u+1)$ in GAN, it follows that as $h_\beta(x) \to -\infty$,

$$f''(e^{h_\beta(x)})e^{2h_\beta(x)} = e^{h_\beta(x)}/\{1 + e^{h_\beta(x)}\} \to 0,$$

and hence the gradient (11) also tends to 0, provided $\nabla_\theta h_\beta(g_\theta(z))$ is bounded. By similar reasoning, we see that the behaviour of $u^2 f''(u)$ as $u \to 0+$ affects the stability of the gradient (11) in the following manner, in the case where the discriminator is confident:

- If $u^2 f''(u) \to 0$ as $u \to 0+$ (e.g., for KL, Hellinger, and GAN), then the gradient (11) tends to vanish;
- If $u^2 f''(u) \to \infty$ as $u \to 0+$ (e.g., for Pearson $\chi^2$), then the gradient (11) tends to explode;
- If $u^2 f''(u) \to 1$ as $u \to 0+$ (e.g., for reverse KL), then the gradient (11) tends to be stable.

Moreover, the reverse KL divergence, denoted as $\mathrm{rKL}(p_*\|p_\theta) = \mathrm{KL}(p_\theta\|p_*)$, has a unique property that $u^2 f''(u)$ is constant for $u > 0$, not just for $u$ near 0, as described by the following result. In general, this property helps to stabilize gradients in $\theta$, by eliminating the variation due to $f''(e^{h_\beta(x)})e^{2h_\beta(x)}$ in (11).

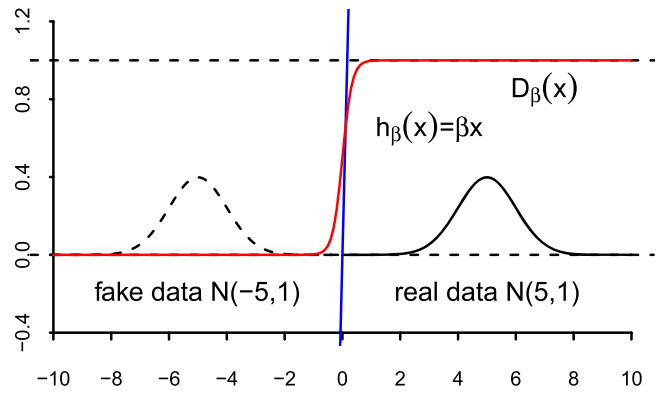
**Proposition 1.** *If $u^2 f''(u) = c$ for $u \in (0, \delta)$, where $\delta > 0$ and $c > 0$ are finite constants, then there exist constants $b_0, b_1 \in \mathbb{R}$ such that $f(u) = b_0 + b_1 u - c\log u$ for $u \in (0, \delta)$, that is, the corresponding f-divergence is equivalent to the reverse KL divergence.*

There are also other $f$-divergences satisfying the condition $u^2 f''(u) \to c$ as $u \to 0+$ for a constant $c > 0$, hence potentially achieving stable gradients when the discriminator is confident. In particular, a simple example is the mixed KL divergence obtained by interpolating the KL and reverse KL divergences:

$$\mathrm{mKL}(p_*\|p_\theta) = (1-\alpha)\,\mathrm{KL}(p_*\|p_\theta) + \alpha\,\mathrm{rKL}(p_*\|p_\theta), \tag{12}$$

with $f(u) = (1-\alpha)u\log u - \alpha\log u$ with $0 < \alpha < 1$. The closer $\alpha$ gets to 0, the more likely the logit $f$GAN algorithm with mKL divergence may suffer vanishing gradients. Another example of divergences that avoid vanishing gradients is associated with GAN2 defined in (4). See Appendix B.3 for further discussion about GAN2 and Theorems 2.5—2.6 in Arjovsky and Bottou (2017).

**TABLE 1** Training a Gaussian model with a logistic discriminator




| Divergence | Gradient in $\theta$ |
|---|---|
| KL | $-\beta E_{p(z)}\{e^{\beta(-5+z)}\}$ |
| Reverse KL | $-\beta$ |
| Hellinger | $-(\beta/2)E_{p(z)}\{e^{\beta(-5+z)/2}\}$ |
| Pearson $\chi^2$ | $-2\beta E_{p(z)}\{e^{-\beta(-5+z)}\}$ |
| GAN | $-\beta E_{p(z)}\{\sigma(\beta(-5+z))\}$ |
| GAN2 | $-\beta E_{p(z)}\{\sigma(-\beta(-5+z))\}$ |

*Note.* Model is $x = \theta + z$ for $z \sim \mathrm{N}(0,1)$. Gradient in $\theta$ (right) is $\nabla_\theta K_f(\theta, \beta)$ in (11) for $f$-divergences or as in Theorem 2.5 of Arjovsky and Bottou (2017) for GAN2, where $\sigma(\cdot)$ is the sigmoid function. GAN: generative adversarial network; KL: Kullback–Liebler.

Table 1 illustrates our analysis in a simple example. The data distribution is $N(5, 1)$ and the generator's distribution is $N(-5, 1)$, which can be well distinguished by a logistic discriminator with $h_\beta(x) = \beta x$ for a large $\beta$ (e.g., $\beta = 10$). The intercept is set to 0 in $h_\beta(x)$, to simplify the gradient calculation. As shown in Table 1, the gradient in $\theta$ from logit *f*GAN either vanishes (for KL, Hellinger, and GAN) or explodes (for Pearson), except for reverse KL and GAN2. In the latter two cases, the location parameter $\theta$ will be updated towards 5 by gradient descent with a proper learning rate. As the process is repeated, the parameter $\theta$ will eventually converge to 5, provided that the discriminator is specified with an intercept, $h_{(\beta_0,\beta_1)}(x) = \beta_0 + \beta_1 x$, and reasonably trained.[1] The gradient in $\theta$ for reverse KL depends on fake data only through the slope $\beta_1$, and hence may fluctuate less than for GAN2 with minibatch sampling. This serves as an indication of gradient stability with reverse KL.

### 4.3.2 | Discriminator gradient

We study the gradient of $K_f(\theta, \beta)$ with respect to $\beta$ for fixed $\theta$, which by using the joint distribution of $(y, x)$ can be expressed as

$$\nabla_\beta K_f(\theta, \beta) = E\left\{f''(e^{h_\beta(x)})e^{h_\beta(x)}(1 + e^{h_\beta(x)})(y - D_\beta(x))\nabla_\beta h_\beta(x)\right\} \tag{13}$$

under the exchange of differentiation and expectation. For the log-likelihood loss (corresponding to GAN), Equation 13 yields the score function in logistic model (3):

$$\nabla_\beta K_f(\theta, \beta) = E\left\{(y - D_\beta(x))\nabla_\beta h_\beta(x)\right\}. \tag{14}$$

For the calibration loss (corresponding to reverse KL) (Tan, 2017), Equation 13 reduces to

$$\nabla_\beta K_f(\theta, \beta) = E\left\{\left(y/D_\beta(x) - 1\right)\nabla_\beta h_\beta(x)\right\}. \tag{15}$$

Setting (15) to 0 can be interpreted as follows: The expectation of $\nabla_\beta h_\beta(x)$ within $\{y = 1\}$, inversely weighted by the probability $D_\beta(x)$, is calibrated to the simple expectation of $\nabla_\beta h_\beta(x)$. If the discriminator is rich enough such that model (3) is correctly specified, then the maximum likelihood estimator is known to achieve the smallest possible asymptotic variance. This is reflected by the fact that the sample version of (14) tends to be well behaved, because $|y - D_\beta(x)|$ is bounded by 1. In contrast, the sample version of (15) may be volatile, because $|y/D_\beta(x) - 1|$ can be very large when $y = 1$ but $D_\beta(x)$ is close to 0. Therefore, given enough capacity of model (3), the log-likelihood loss seems to be more desirable than the calibration loss (corresponding to reverse KL) for updating $\beta$.

## 4.4 | Calibrated GAN

From Section 4.3, we see that there are competing considerations in choosing a divergence measure in logit *f*GAN. On one hand, it seems suitable to use the reverse KL divergence or mixed KL for achieving stable gradients in updating the generator parameters $\theta$. On the other hand, using the log-likelihood loss (as in GAN) seems desirable for efficient estimation of $\beta$ in the discriminator. To take advantage of ("calibrate") both choices, consider the following program, called calibrated GAN (Cal-GAN):

$$\begin{cases} \max\limits_{\beta} \; K_{\mathrm{GAN}}(\theta, \beta) \quad \text{with } \theta \text{ fixed}, & \text{(16a)} \\ \min\limits_{\theta} \; K_f(\theta, \beta) \quad \text{with } \beta \text{ fixed}, & \text{(16b)} \end{cases}$$

where $K_{\mathrm{GAN}}(\theta, \beta)$ is from (2) and $K_f(\theta, \beta)$ in (7) can be defined according to any *f*-divergence. Our recommendation is to use the reverse KL divergence, for which $K_f(\theta, \beta)$ becomes

$$K_{\mathrm{rKL}}(\theta, \beta) = E_{p_*(x)}\left\{-e^{-h_\beta(x)}\right\} - E_{p_\theta(x)}\left\{h_\beta(x)\right\} + 1, \tag{17}$$

or the mixed KL divergence in (12), for which $K_f(\theta, \beta)$ is provided in Appendix A. The stationary condition associated with program (16) is $\nabla_\beta K_{\mathrm{GAN}}(\theta, \beta) = 0$ and $\nabla_\theta K_f(\theta, \beta) = 0$. Algorithm 2 presents a procedure for solving (16), based on stochastic gradient descent.

**Algorithm 2** Calibrated GAN with *f*-divergence

**repeat**

**Sampling:** Same as in Algorithm 1.

**Updating:** Denote $\hat{K}_f(\theta, \beta)$ as in Algorithm 1, and

$$\hat{K}_{\mathrm{GAN}}(\theta, \beta) = \frac{1}{m}\sum_{i=1}^{m} \log\{D_\beta(x_i)\} + \frac{1}{m}\sum_{i=1}^{m} \log\left\{1 - D_\beta(g_\theta(z_i))\right\};$$

Update $\beta$ by ascending the gradient $\nabla_\beta \hat{K}_{\mathrm{GAN}}(\theta, \beta)$; Update $\theta$ by descending the gradient $\nabla_\theta \hat{K}_f(\theta, \beta)$.

**until** convergence

[1] As discussed next, training of the discriminator may not be desirable in logit *f*GAN with reverse KL, which points to the development of Cal-GAN with two objectives in Section 4.4.

We provide several comments. First, program (10) is a saddle-point problem with a single objective, whereas (16) involves two objectives coupled with each other. Nevertheless, we show that if the discriminator is sufficiently rich and well trained, then program (16) also leads to minimization of the $f$-divergence $\mathcal{D}_f(p_*\|p_\theta)$, similarly as the $f$GAN program (10). See Appendix B for further discussion about the various adversarial algorithms with two objectives.

**Proposition 2.** *For any fixed $\theta$, suppose that the optimal discriminator is obtained from 16a, with $\beta_\theta^*$ such that $h_{\beta_\theta^*}(x) = \log\{p_*(x)/p_\theta(x)\}$. Then $\{\nabla_\theta K_f(\theta,\beta)\}|_{\beta=\beta_\theta^*} = \nabla_\theta \mathcal{D}_f(p_*\|p_\theta)$.*

The condition on $\beta_\theta^*$ is automatically satisfied if the discriminator is nonparametric. The nonparametric (or infinite capacity) condition is imposed in, as far as we know, all GAN-related theoretical results comparable with our Proposition 2. See, for example, Goodfellow et al. (2014, Section 4) and Zhao et al. (2017, Section 2.2). The condition that the optimal discriminator is used can also be found in Theorem 2.5 in Arjovsky and Bottou (2017). Further research is needed to deal with the use of discriminators of limited capacity.

Second, Cal-GAN can be interpreted from two complementary angles. On one hand, Cal-GAN is a modification of logit $f$GAN with the objective replaced by maximum likelihood in training the discriminator. On the other hand, Cal-GAN can also be regarded as a modification of GAN with the objective replaced by the logit $f$GAN objective, for example, corresponding to reverse KL, in training the generator. This modification is of a similar nature as the log$D$ trick to overcome vanishing gradients but leads to minimization of the reverse KL divergence, instead of the objective underlying GAN2, which seems problematic with a negative JS term (Arjovsky & Bottou, 2017, Theorem 2.5).

Third, the use of two objectives in (16) for updating $\beta$ and $\theta$ allows Cal-GAN to be stated in seemingly different ways with some caveat. With $K_f(\theta,\beta)$ based on reverse KL, program (16), $\max_\beta K_{\text{GAN}}(\theta,\beta)$ with $\theta$ fixed and $\min_\theta K_{\text{rKL}}(\theta,\beta)$ with $\beta$ fixed, is equivalent to both (1) and

$$\begin{cases} \max\limits_{\beta} \; K_{\text{GAN}}(\theta,\beta) \quad \text{with } \theta \text{ fixed}, & \text{(18a)} \\ \min\limits_{\theta} \; E_{p_*(x)}\left\{h_\beta(x)\right\} - E_{p_\theta(x)}\left\{h_\beta(x)\right\} \quad \text{with } \beta \text{ fixed}, & \text{(18b)} \end{cases}$$

because for fixed $\beta$, minimization of $K_{\text{rKL}}(\theta,\beta)$ is equivalent to that of $E_{p_*(x)}\{h_\beta(x)\} - E_{p_\theta(x)}\{h_\beta(x)\}$ and that of $-E_{p_\theta(x)}\{h_\beta(x)\}$ over $\theta$. Therefore, program (18) leads to minimization of the reverse KL divergence for learning $\theta$ provided the discriminator is sufficiently rich and well trained. A method defined as (18) was mentioned in Section 4.2 of Chen et al. (2018) but was incorrectly claimed to result in minimization of the symmetric KL divergence, that is, mKL$(p_*, p_\theta)$ in (12) with $\alpha = 0.5$.[2] Moreover, minimization of symmetric KL for learning $\theta$ can be realized via program (16) with $K_f(\theta,\beta)$ based on symmetric KL, as described in Appendix A.

## 5 | EXPERIMENTS

We conduct several experiments to study the performance of our method (Cal-GAN) and various existing methods, with both visual and numerical evaluation. See Supporting Information for experimental details and additional results including generated samples.

### 5.1 | Gaussian mixture model synthetic data

The synthetic data consist of 1,600 training examples generated from a 2D Gaussian mixture model with 32 equally weighted, low-variance ($\sigma = 0.1$) Gaussian components, uniformly laid out on four concentric circles as in Figure S2(a) in the Supporting Information. The data distribution exhibits many modes separated by low-probability regions, which makes it suitable to examine how well different learning methods can deal with multiple modes. We experiment with three learning methods: GAN2, logit-$f$GAN with mKL (Algorithm 1), and Cal-GAN with mKL (Algorithm 2), with $\alpha$ from $\{0.25, 0.5, 0.75, 1\}$. The method with $\alpha = 0$ is known to suffer from gradient vanishing (Section 4.3).

Table 2 reports the ''covered modes'' and ''realistic ratio'' as numerical measures of how well the multimodal data are fitted, similarly as in Dumoulin et al. (2017). See Figure S2 in the Supporting Information for visual comparison of generated samples. The main observations are as follows. First, Cal-GAN performs better than the corresponding logit-$f$GAN with considerable margins in both quality measures, across different values of $\alpha$. Second, the use of mixed KL with a larger $\alpha$ appears better at generating ''realistic'' samples but tends to drop modes, which is consistent with related discussion (Goodfellow, 2017; Huszár, 2015; Theis et al., 2016). Potentially, a balance of ''covered modes'' and ''realistic ratio'' can be achieved by tuning the choice of $\alpha$. Third, GAN2 appears good at generating ''realistic'' samples but suffers substantial mode dropping, and the overall performance is worse than Cal-GAN with reverse KL. In summary, this experiment demonstrates the superiority of Cal-GAN over GAN2 and logit-$f$GAN in both covering modes and generating realistic samples.

[2]The optimal $h_\beta(x)$ from (18a) is $h_\beta(x) = \log\{p_*(x)/p_\theta(x)\}$ given enough model capacity and substituting this $h_\beta(x)$ into the objective in (18b) gives $E_{p_*(x)}[\log\{p_*(x)/p_\theta(x)\}] - E_{p_\theta(x)}[\log\{p_*(x)/p_\theta(x)\}]$. However, in (18b), the first term $E_{p_*(x)}[\log\{p_*(x)/p_\theta(x)\}]$ is treated as $E_{p_*(x)}\{h_\beta(x)\}$ (fixed), and only the second term $-E_{p_\theta(x)}[\log\{p_*(x)/p_\theta(x)\}]$ is minimized over $\theta$. See Appendix B.2 for further discussion.

**TABLE 2** Numerical evaluations with Gaussian mixture model (32 components) synthetic data

| Methods | Covered modes | Realistic ratio |
|---|---|---|
| GAN2 | 23.16 ± 1.32 | 0.908 ± 0.008 |
| logit-*f*GAN | | |
| mKL ($\alpha = 0.25$) | 28.23 ± 0.97 | 0.793 ± 0.019 |
| mKL ($\alpha = 0.5$) | 27.36 ± 0.93 | 0.814 ± 0.018 |
| mKL ($\alpha = 0.75$) | 25.97 ± 1.02 | 0.821 ± 0.019 |
| rKL (mKL $\alpha = 1.0$) | 25.05 ± 1.13 | 0.867 ± 0.015 |
| Cal-GAN | | |
| mKL ($\alpha = 0.25$) | 28.31 ± 0.91 | 0.866 ± 0.017 |
| mKL ($\alpha = 0.5$) | 27.41 ± 0.97 | 0.878 ± 0.014 |
| mKL ($\alpha = 0.75$) | 26.87 ± 1.04 | 0.893 ± 0.010 |
| rKL (mKL $\alpha = 1.0$) | 26.04 ± 1.01 | 0.905 ± 0.011 |

*Note*. A mode is defined to be covered when there exist generated samples close to the mode within a threshold. The "covered modes" metric is defined as the number of covered modes by a set of generated samples. The "realistic ratio" metric is defined as the proportion of generated samples that are close to a mode. The measurement details are presented in the Supporting Information. Mean and *SD* are from 10 independent runs. GAN: generative adversarial network; KL: Kullback–Liebler.

## 5.2 | MNIST

MNIST consists of 60,000 training examples and 10,000 test examples, where each example is a handwritten digit image of size $28 \times 28$. In addition to visual inspection, we evaluate sample quality using the inception score (Salimans et al., 2016), which is defined by taking into account that samples should be both diverse and realistic. We employ a well-trained classification network from Chen et al. (2018) to calculate the inception scores.

Figure 2a shows the inception scores for different methods during training. Compared with logit-*f*GAN, DCGAN (Radford, Metz, & Chintala, 2016), and WGAN-GP (Gulrajani, Ahmed, Arjovsky, Dumoulin, & Courville, 2017), Cal-GAN converges faster and yields consistently better inception scores over the course of training. It is found in Arjovsky and Bottou (2017) that DCGAN (GAN2) suffers from large variability of the generator gradient as the discriminator is being trained. For a similar experiment, Figure 2b shows that Cal-GAN achieves more stable gradients for the generator than DCGAN. This confirms that calibration of the training objective for the generator in Cal-GAN is effective in improving stability.

## 5.3 | CIFAR-10

CIFAR-10 consists of 50,000 training examples and 10,000 examples, where each example is a natural image of size $3 \times 32 \times 32$ from 10 classes. CIFAR-10 is more complex than MNIST and widely adopted for experiments on image generation. We employ the standard CNN in Miyato et al. (2018) and compare various methods including the hinge GAN (Lim & Ye, 2017). Each method is implemented using batch normalization on the generator (Ioffe & Szegedy, 2015), but two different regularization techniques on the discriminator, either batch normalization or spectral normalization (Miyato et al., 2018). The latter technique has recently been shown to yield outstanding results on CIFAR-10, especially when used with the hinge GAN (Lim & Ye, 2017).

Table 3 reports both inception scores (Salimans et al., 2016) and Frechet inception distances (Heusel, Ramsauer, Unterthiner, Nessler, & Hochreiter, 2017). Figure 3 shows the inception scores over training iterations, and Figure S6 in the Supporting Information shows the corresponding plots of Frechet inception distances. For Cal-GAN, the results with mixed KL are similar to those with the reverse KL and hence not reported. For logit-*f*GAN with mixed KL ($\alpha < 1$), we find that the training often crashes due to unstable gradients, and no reasonable result is obtained.

From Table 3 and, over the course of training, Figure 3, we see that Cal-GAN outperforms DCGAN, WGAN-GP, and logit-*f*GAN consistently, sometimes with large margins, when either batch or spectral normalization is used on the discriminator. The improvement is more substantial in the former case, indicating that the capability of Cal-GAN depends less on specialized training techniques. Although Cal-GAN also outperforms hinge GAN with batch normalization, the two methods achieve similar results when spectral normalization is used.

It should be mentioned that better generation results have been obtained on CIFAR-10 with more complex ResNet (He et al., 2016) than the standard CNN. Further experiments using Cal-GAN in those settings can be pursued in future work.

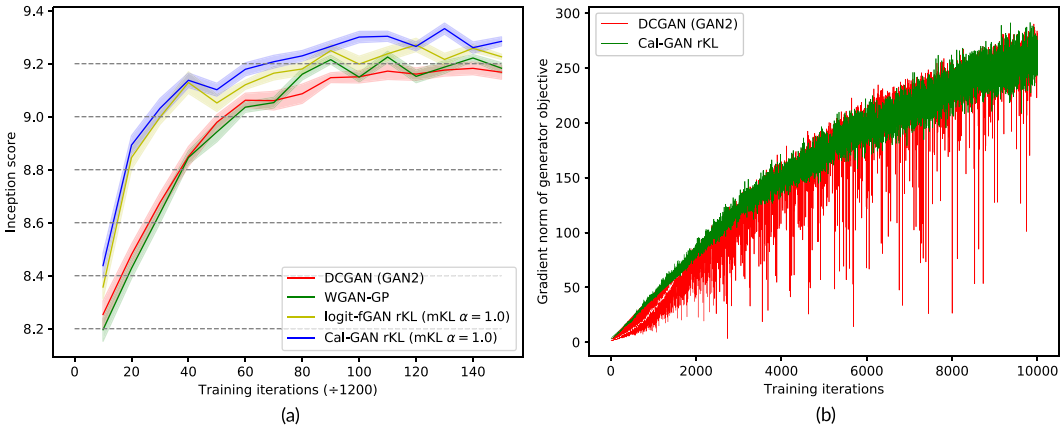



**FIGURE 2** (a) Inception scores during training for different methods on MNIST. Each solid line gives the mean, and the shaded area gives the standard deviation, based on 10 independent runs. See Supporting Information for results with mixed KL ($\alpha < 1$). (b) Comparison of generator gradients between DCGAN and Cal-GAN on MNIST. We first trained models for 10 epochs. Then, with the generator fixed, we train a discriminator from scratch and measure the gradient norm for the generator, $\|\nabla_\theta K_{\mathrm{rKL}}(\theta, \beta)\|$, similarly as in Arjovsky and Bottou (2017). The variability of the generator gradients for DCGAN is much larger than Cal-GAN. See Supporting Information for comparison between WGAN-GP or logit-*f*GAN and Cal-GAN

**TABLE 3** Inception scores and FIDs on CIFAR-10, based on the standard CNN used in Miyato et al. (2018)

| Methods | Inception score | FID | Inception score | FID |
|---|---|---|---|---|
| | Batch norm on discriminator | | Spectral norm on discriminator | |
| DCGAN (Radford et al., 2016) (GAN2) | 6.39 ± 0.13 | 52.8 ± 1.30 | 7.31 ± 0.09 | 32.2 ± 0.69 |
| WGAN-GP (Gulrajani et al., 2017) | 6.80 ± 0.15 | 39.9 ± 1.13 | 7.36 ± 0.12 | 31.7 ± 0.85 |
| Hinge GAN (Lim & Ye, 2017) | 6.94 ± 0.11 | 37.6 ± 0.93 | 7.58 ± 0.10 | 27.0 ± 0.47 |
| logit-*f*GAN rKL | 6.57 ± 0.12 | 46.8 ± 0.99 | 7.46 ± 0.12 | 28.9 ± 0.55 |
| Cal-GAN rKL | 6.99 ± 0.07 | 36.6 ± 0.67 | 7.54 ± 0.08 | 28.1 ± 0.38 |

*Note.* Results for all methods are from our own implementation, including DCGAN, WGAN-GP, and hinge GAN, for which the released code in the original work is reproduced if possible. The inception scores and FIDs are calculated using the classification model from Salimans et al. (2016). Mean and *SD* are from 10 independent runs. GAN: generative adversarial network; FID: Frechet inception distance.

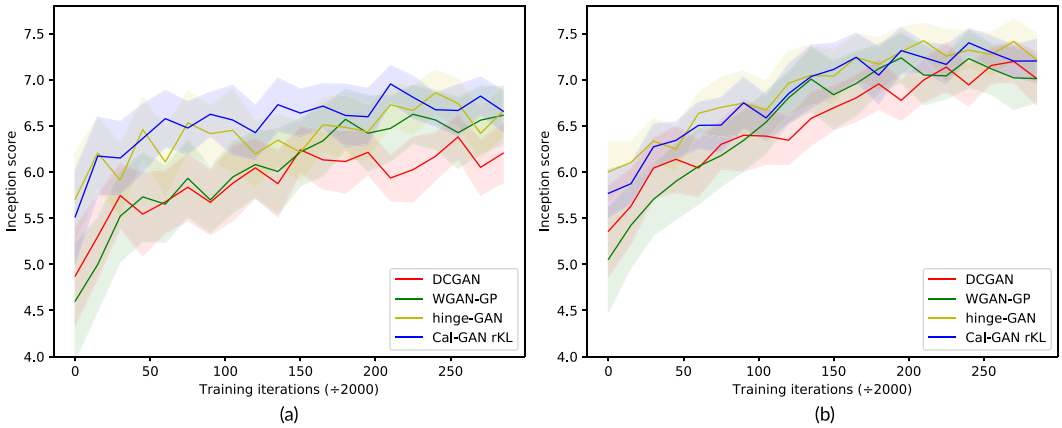



**FIGURE 3** Inception scores during training on CIFAR-10 when (a) batch normalization or (b) spectral normalization is used on the discriminator. Each solid line gives the mean, and the shaded area represents the standard deviation, based on 10 independent runs. See Figure S6 in the Supporting Information for the corresponding plots of Frechet inception distances

## 6 | CONCLUSION

We provide both theoretical analysis and empirical results in support of the use of Cal-GAN. Further comparison is desirable between Cal-GAN and alternative GAN methods (Arjovsky et al., 2017; Li, Swersky, & Zemel, 2015; Lim & Ye, 2017; Zhao et al., 2017) with different network architectures and image datasets. On the other hand, Cal-GAN can be complementary and useful to a broad range of GAN-related work,

including feature matching (Salimans et al., 2016; Warde-Farley & Bengio, 2017), inference modelling and auto-encoding (Dumoulin et al., 2017; Mescheder, Nowozin, & Geiger, 2017), semisupervised learning (Gan et al., 2017; Salimans et al., 2016), and others. Combination and extension of these ideas are promising directions for future work.

## DATA AVAILABILITY STATEMENT

Computer codes are available from the corresponding authors upon request.

### ORCID

*Zhiqiang Tan* https://orcid.org/0000-0003-1780-6839
*Zhijian Ou* https://orcid.org/0000-0002-9018-5074

### REFERENCES

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 28*, 131–142.

Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *ICLR 2017*.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proceedings of Machine Learning Research, 70*, 214–223.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Chen, L., Dai, S., Pu, Y., Zhou, E., Li, C., Su, Q., ..., & Carin, L. (2018). Symmetric variational autoencoder and connections to adversarial learning. In *Proceedings of Machine Learning Research*, *84*, 661–669.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., & Courville, A. (2017). Adversarially learned inference. In *ICLR 2017*.

Gan, Z., Chen, L., Wang, W., Pu, Y., Zhang, Y., Liu, H., ..., & Carin, L. (2017). *Triangle generative adversarial networks*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 5247–5256). Curran Associates, Inc..

Goodfellow, I. J. (2017). NIPS 2016 tutorial: Generative adversarial networks. arXiv:1701.00160.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ..., & Bengio, Y. (2014). *Generative adversarial nets*. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27* (pp. 2672–2680). Curran Associates, Inc..

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). *Improved training of Wasserstein GANs*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 5767–5777). Curran Associates, Inc..

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). *GANs trained by a two time-scale update rule converge to a local Nash equilibrium*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 6626–6637). Curran Associates, Inc..

Huszár, F. (2015). How (not) to train your generative model: Scheduled sampling, likelihood, adversary? arXiv:1511.05101.

Huszár, F. (2016). An alternative update rule for generative adversarial networks. blogpost. http://www.inference.vc/an-alternative-update-rule-for-generative-adversarial-networks/

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of Machine Learning Research*, *37*, 448–456.

Li, Y., Swersky, K., & Zemel, R. S. (2015). Generative moment matching networks. In *Proceedings of Machine Learning Research*, *37*, 1718–1727.

Lim, J. H., & Ye, J. C. (2017). Geometric GAN. arXiv:1705.02894.

Lin, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery*, *6*, 259–275.

Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of Machine Learning Research*, *70*, 2391–2400.

Minka, T. (2005). Divergence measures and message passing, *Microsoft Technical Report (MSR-TR-2005-173)*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *ICLR 2018*.

Mohamed, S., & Lakshminarayanan, B. (2017). Learning in implicit generative models. arXiv:1610.03483v4.

Nguyen, T., Le, T., Vu, H., & Phung, D. (2017). *Dual discriminator generative adversarial nets*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 2670–2680). Curran Associates, Inc..

Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, *56*, 5847–5861.

Nowozin, S., Cseke, B., & Tomioka, R. (2016). *f-GAN: Training generative neural samplers using variational divergence minimization*. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 29* (pp. 271–279). Curran Associates, Inc..

Poole, B., Alemi, A., Sohl-Dickstein, J., & Angelova, A. (2016). Improved generator objectives for GANs. In *NIPS Workshop on Adversarial Training*.

Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR 2016*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). *Improved techniques for training GANs*. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 29* (pp. 2234–2242). Curran Associates, Inc..

Tan, Z. (2017). Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data. arXiv:1710.08074.

Theis, L., Den Oord, A. V., & Bethge, M. (2016). A note on the evaluation of generative models. In *ICLR 2016*.

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., & Matsuo, Y. (2016). Generative adversarial nets from a density ratio estimation perspective. arXiv:1610.02920v2.

Warde-Farley, D., & Bengio, Y. (2017). Improving generative adversarial networks with denoising feature matching. In *ICLR 2017*.

Zhao, J., Mathieu, M., & LeCun, Y. (2017). Energy-based generative adversarial networks. In *ICLR 2017*.

## SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

**How to cite this article:** Tan Z, Song Y, Ou Z. Calibrated adversarial algorithms for generative modelling. *Stat*. 2019;8:e224. https://doi.org/10.1002/sta4.224

## APPENDIX A: MIXED KL DIVERGENCE

For the mixed KL divergence in (12), the objective $K_f(\theta, \beta)$ in (7) becomes

$$K_{\mathrm{mKL}}(\theta, \beta) = E_{p_*(x)}\left\{-\alpha e^{-h_\beta(x)} + (1-\alpha)h_\beta(x)\right\} - E_{p_\theta(x)}\left\{\alpha h_\beta(x) + (1-\alpha)e^{h_\beta(x)}\right\} + 1. \tag{A1}$$

Then Cal-GAN program (16), $\max_\beta K_{\mathrm{GAN}}(\theta, \beta)$ and $\min_\theta K_{\mathrm{mKL}}(\theta, \beta)$, is equivalent to

$$\begin{cases} \max\limits_{\beta} K_{\mathrm{GAN}}(\theta, \beta) \quad \text{with } \theta \text{ fixed}, & \text{(A2a)} \\ \min\limits_{\theta} -E_{p_\theta(x)}\left\{\alpha h_\beta(x) + (1-\alpha)e^{h_\beta(x)}\right\} \quad \text{with } \beta \text{ fixed}, & \text{(A2b)} \end{cases}$$

and hence also to

$$\begin{cases} \max\limits_{\beta} K_{\mathrm{GAN}}(\theta, \beta) \quad \text{with } \theta \text{ fixed}, & \text{(A3a)} \\ \min\limits_{\theta} E_{p_*(x)}\left\{\alpha h_\beta(x) + (1-\alpha)e^{h_\beta(x)}\right\} - E_{p_\theta(x)}\left\{\alpha h_\beta(x) + (1-\alpha)e^{h_\beta(x)}\right\} \quad \text{with } \beta \text{ fixed}. & \text{(A3b)} \end{cases}$$

In (A3b), the first term $E_{p_*(x)}\{\alpha h_\beta(x) + (1-\alpha)e^{h_\beta(x)}\}$, free of $\theta$, can be removed or modified in another way (as long as free of $\theta$) without affecting the result of the program. Compared with (18b) in the paper, the objective in (A2b) or (A3b) involves the expectations of both $h_\beta(x)$ and $e^{h_\beta(x)}$ under $p_\theta(x)$, which is crucial for the program to achieve minimization of the mixed KL divergence.

We show that D2GAN in Nguyen et al. (2017) can be seen as an extension of *f*GAN with mixed KL to use two discriminators. By parameterizing $D_1(x) = a e^{h_{\beta_1}(x)}$, $D_2(x) = b e^{-h_{\beta_2}(x)}$ and relabelling $(\alpha, \beta) = (a, b)$ in Equation (1) of Nguyen et al. (2017),[3] this method can be restated as

$$\min_\theta \max_{\beta_1, \beta_2} K_{\mathrm{D2GAN}}(\theta, \beta_1, \beta_2) = E_{p_*(x)}\left\{-\alpha e^{-h_{\beta_2}(x)} + (1-\alpha)h_{\beta_1}(x)\right\} - E_{p_\theta(x)}\left\{\alpha h_{\beta_2}(x) + (1-\alpha)e^{h_{\beta_1}(x)}\right\}, \tag{A4}$$

where $\alpha = b/(a+b)$ and $h_{\beta_1}(x)$ and $h_{\beta_2}(x)$ are two representations of the log density ratio $\log\{p_*(x)/p_\theta(x)\}$. The objective in (A4) can be obtained from $K_{\mathrm{mKL}}(\theta, \beta)$ in (A1), by resetting $h_\beta(x)$ to either $h_{\beta_1}(x)$ or $h_{\beta_2}(x)$ for two discriminators. In fact, $h_\beta(x)$ is reset according to the location of $h_\beta(x)$ in the decomposition: $K_{\mathrm{mKL}}(\theta, \beta) = (1-\alpha)K_{\mathrm{KL}}(\theta, \beta) + \alpha K_{\mathrm{rKL}}(\theta, \beta)$, where $K_{\mathrm{KL}}(\theta, \beta)$ is $K_f(\theta, \beta)$ based on KL, and $K_{\mathrm{rKL}}(\theta, \beta)$ is based on reverse KL as in (17).

## APPENDIX B: UNDERSTANDING ADVERSARIAL ALGORITHMS

We study several adversarial algorithms with two objectives. To focus on the main issues, we treat the saddle-point problem (10) for logit *f*GAN interchangeably with

$$\begin{cases} \max\limits_{\beta} K_f(\theta, \beta) \quad \text{with } \theta \text{ fixed}, & \text{(B1a)} \\ \min\limits_{\theta} K_f(\theta, \beta) \quad \text{with } \beta \text{ fixed}. & \text{(B1b)} \end{cases}$$

In general, Algorithm 1 based on gradient descent is expected to find a solution satisfying the stationary condition $\nabla_\beta K_f(\theta, \beta) = 0$ and $\nabla_\theta K_f(\theta, \beta) = 0$ for program (B1).

[3]The relabelling of $(\alpha, \beta)$ to $(a, b)$ is to avoid confusion with $\alpha$ in (A4). The parametrization of $D_1(x)$ and $D_2(x)$ is consistent with the fact that $D_1(x)$ and $D_2(x)$ are positive and unbounded in Nguyen et al. (2017). For the softplus activation used to output $D_1(x)$ and $D_2(x)$ in Nguyen et al. (2017), the functions before the activation cannot be interpreted as log density ratios.

## B.1 | Algorithms related to *f*GAN

In addition to program (B1) corresponding to logit *f*GAN, there are two other programs suggested in Section 3.4, Uehara et al. (2016):

$$\begin{cases} \max_{\beta} K_f(\theta, \beta) \quad \text{with } \theta \text{ fixed}, & \text{(B2a)} \\ \min_{\theta} K_f^{(1)}(\theta, \beta) = -E_{p_\theta(x)}\{f'(U_\beta(x))\} \quad \text{with } \beta \text{ fixed}, & \text{(B2b)} \end{cases}$$

and

$$\begin{cases} \max_{\beta} K_f(\theta, \beta) \quad \text{with } \theta \text{ fixed}, & \text{(B3a)} \\ \min_{\theta} K_f^{(2)}(\theta, \beta) = E_{p_\theta(x)}\{f(U_\beta(x))\} \quad \text{with } \beta \text{ fixed}. & \text{(B3b)} \end{cases}$$

Program (B2) was also discussed in Nowozin et al. (2016) as a generalization of the log*D* trick to *f*GAN. Moreover, there is another program suggested in Equation (23), Mohamed and Lakshminarayanan (2017):

$$\begin{cases} \max_{\beta} K_f(\theta, \beta) \quad \text{with} \theta \text{ fixed}, & \text{(B4a)} \\ \min_{\theta} K_f^{(3)}(\theta, \beta) = E_{p_\theta(x)}\{U_\beta(x) f'(U_\beta(x))\} \quad \text{with } \beta \text{ fixed}. & \text{(B4b)} \end{cases}$$

Compared with program (B1), each of the three programs (B2)–(B4) uses an objective function different from $K_f(\theta, \beta)$ for training the generator, whereas the Cal-GAN program (16) replaces the objective $K_f(\theta, \beta)$ by maximum likelihood for training the discriminator. We show that, unlike program (16) or (B1), none of the programs (B2)–(B4) in general leads to minimization of the *f*-divergence $\mathcal{D}_f(p_*\|p_\theta)$ even with a nonparametric discriminator.

**Proposition 3.** *For any fixed $\theta$, suppose that $\beta_\theta^*$ is obtained from (B1a) such that $h_{\beta_\theta^*}(x) = \log\{p_*(x)/p_\theta(x)\}$. Then the following results hold.*

*(i)*

$$\left\{\nabla_\theta K_f^{(1)}(\theta, \beta)\right\}\Big|_{\beta=\beta_\theta^*} = \nabla_\theta \mathcal{D}_{\tilde{f}}(p_*\|p_\theta), \tag{B5}$$

*where $K_f^{(1)}(\theta, \beta)$ is from (B2b), and $\tilde{f} : \mathbb{R}_+ \to \mathbb{R}$ is a convex function satisfying $\tilde{f}''(u) = f''(u)/u$.*

*(ii) In general,*

$$\left\{\nabla_\theta K_f^{(2)}(\theta, \beta)\right\}\Big|_{\beta=\beta_\theta^*} \neq \nabla_\theta \mathcal{D}_f(p_*\|p_\theta), \tag{B6}$$

$$\left\{\nabla_\theta K_f^{(3)}(\theta, \beta)\right\}\Big|_{\beta=\beta_\theta^*} \neq \nabla_\theta \mathcal{D}_f(p_*\|p_\theta), \tag{B7}$$

*where $K_f^{(2)}(\theta, \beta)$ is from (B3b) and $K_f^{(3)}(\theta, \beta)$ from (B4b).*

Proposition 3(i) implies that if the optimal discriminator is obtained but model $p_\theta(x)$ may be misspecified, then program (B2a), in general, minimizes not the divergence $\mathcal{D}_f(p_*\|p_\theta)$ but $\mathcal{D}_{\tilde{f}}(p_*\|p_\theta)$ for another convex function $\tilde{f}$. This result extends Theorem 2 in Nowozin et al. (2016), which only shows that programs (B1a) and (B2a) lead to a stationary point $\theta_0$ satisfying $p_*(x) \equiv p_{\theta_0}(x)$, that is, model $p_\theta(x)$ is well specified. Moreover, application of our result to GAN2 gives a sharper conclusion than Theorem 2.5 in Arjovsky and Bottou (2017) (see Proposition 4).

Proposition 3(ii) shows that if the optimal discriminator is obtained but model $p_\theta(x)$ may be misspecified, program (B3a) or (B4a), in general, does not admit a stationary point that is a minimizer of $\mathcal{D}_f(p_*\|p_\theta)$. Nevertheless, as a special case, both programs (B3a) and (B4a) can be shown to yield $\theta_0$ satisfying $p_*(x) \equiv p_{\theta_0}(x)$ as a stationary point when model $p_\theta(x)$ is well specified. The "misbehaviour" of (B3a) or (B4a) can be explained similarly as in the discussion about program (B8) in Section B.2.

## B.2 | Algorithms related to Cal-GAN

Compared with Cal-GAN, it seems tempting to consider the following program

$$\begin{cases} \max_{\beta} K_{\mathrm{GAN}}(\theta, \beta) \quad \text{with } \theta \text{ fixed}, & \text{(B8a)} \\ \min_{\theta} E_{p_\theta(x)}\{f(U_\beta(x))\} \quad \text{with } \beta \text{ fixed}. & \text{(B8b)} \end{cases}$$

A possible reasoning that if $\beta$ is set to $\beta_\theta^*$ such that $U_{\beta_\theta^*}(x) = p_*(x)/p_\theta(x)$, then the objective in (B8b) reduces to $\mathcal{D}_f(p_*\|p_\theta)$, which is to be minimized. However, somewhat surprisingly, program (B8) does not in general lead to minimization of $\mathcal{D}_f(p_*\|p_\theta)$. In fact, the gradient for minimization of $E_{p_\theta(x)}\{f(U_\beta(x))\}$ over $\theta$ with fixed $\beta = \beta_\theta^*$ is

$$\left[\nabla_\theta E_{p_\theta(x)}\{f(U_\beta(x))\}\right]|_{\beta=\beta_\theta^*} = \int \{\nabla_\theta p_\theta(x)\} f\left(\frac{p_*(\theta)}{p_\theta(x)}\right) \mathrm{d}x, \tag{B9}$$

which in general differs from $\nabla_\theta \mathcal{D}_f(p_*\|p_\theta)$, as shown in (S4) in the Supporting Information. That is, (B9) does not account for differentiation of $f(U_{\beta_\theta^*}(x))$ with respect to $\theta$. With (B8b) replaced by (18b), this discussion also explains why program (18) does not lead to minimization of symmetric KL as claimed in Chen et al. (2018).

Program (B8) can be rectified as follows, to achieve minimization of $\mathcal{D}_f(p_*\|p_\theta)$:

$$\min_\theta E_{p_\theta(x)}\{f(e^{h_{\beta_\theta}(x)})\}, \tag{B10}$$

$$\text{where} \quad \beta_\theta = \text{argmax}_\beta K_{\text{GAN}}(\theta,\beta). \tag{B11}$$

A difficulty with this program is that the gradient of the objective over $\theta$ in (B10) is in general numerically intractable because $\beta_\theta$ is implicitly defined through (B11).

The Cal-GAN program (16) differs from (B8), with $K_f(\theta,\beta)$ in place of $E_{p_\theta(x)}\{f(U_\beta(x))\}$. From Proposition 2, we see that given enough capacity and training of the discriminator, (16) leads to minimization of $\mathcal{D}_f(p_*\|p_\theta)$, because the objective $K_f(\theta,\beta)$ not only satisfies $K_f(\theta,\beta)|_{\beta=\beta_\theta^*} = \mathcal{D}_f(p_*\|p_\theta)$ but also $\{\nabla_\theta K_f(\theta,\beta)\}|_{\beta=\beta_\theta^*} = \nabla_\theta \mathcal{D}_f(p_*\|p_\theta)$.

## B.3 | GAN with log*D* trick

To avoid vanishing gradients, GAN is often used with the log*D* trick, that is, GAN2 defined in (4). Theorem 2.5 in Arjovsky and Bottou (2017) shows that if the discriminator is sufficiently rich and well trained, then GAN2 effectively minimizes

$$\text{GAN2}(p_*,p_\theta) = \text{rKL}(p_*\|p_\theta) - 2\text{JS}(p_*\|p_\theta), \tag{B12}$$

which seems to be an extrapolation of the reverse KL away from the JS divergence. Although the difference between two divergences may not be a proper divergence, application of Proposition 3(i) reveals that the objective for GAN2 as stated above is actually an *f*-divergence for a convex function *f* such that $f''(u) = f''_{\text{JS}}(u)/u = 1/\{u^2(u+1)\}$, where $f_{\text{JS}}(u) = u\log u - (u+1)\log(u+1)$.

**Proposition 4.** *GAN2*$(p_*,p_\theta)$ *is an f-divergence with* $f(u) = (u+1)\log\{(u+1)/u\}$, *which is convex with* $f_{JS}(u) = u\log u - (u+1)\log(u+1)$.

Because $\text{JS}(p_*\|p_\theta)$ is symmetric in $p_*$ and $p_\theta$, this result suggests, as similarly discussed in Arjovsky and Bottou (2017), that GAN2 is at least as prune to mode dropping as implied by the reverse KL. But it also explains why GAN2 training is theoretically consistent, although the negative JS divergence is "pushing for the distributions to be different, which seems like a fault" (Arjovsky & Bottou, 2017).

For GAN2, the gradient for updating $\theta$ is

$$-E_{p(z)}\left\{(1-D_\beta(x))|_{x=g_\theta(z)}\nabla_\theta h_\beta(g_\theta(z))\right\}, \tag{B13}$$

by direct calculation using $D_\beta(x) = \sigma(h_\beta(x))$. With logistic discriminator $D_\beta(x)$, Theorem 2.6 in Arjovsky and Bottou (2017) about gradient fluctuation in GAN2 seems inapplicable, because $\{\nabla_x D_\beta(x)\}/D_\beta(x) = \{1 - D_\beta(x)\}\nabla_x h_\beta(x)$ may be far from being Cauchy. Moreover, the gradient (B13) may remain nonnegligible and stable, similarly as $-E_{p(z)}\{\nabla_\theta h_\beta(g_\theta(z))\}$, the gradient in $\theta$ with the reverse KL, when the fake and real data are well separated such that $D_\beta(x) \to 0$ and $h_\beta(x) \to -\infty$ for $x = g_\theta(z)$. As discussed in Section 4.3, such stability of (B13) can also be seen from the fact that $u^2 f''(u) \to 1$ as $u \to 0+$ with $f(\cdot)$ in Proposition 4.

## B.4 | Hinge GAN

The population version of hinge GAN is (Lim & Ye, 2017; Miyato et al., 2018)

$$\begin{cases} \max_\beta K_{\text{hinge}}(\theta,\beta) & \text{with } \theta \text{ fixed}, \quad \text{(B14a)} \\ \min_\theta -E_{p_\theta(x)}\left\{H_\beta(x)\right\} & \text{with } \beta \text{ fixed}, \quad \text{(B14b)} \end{cases}$$

with the hinge loss $K_{\text{hinge}}(\theta,\beta) = E_{p_*(x)}\{\max(0,1-H_\beta(x))\} + E_{p_\theta(x)}\{\max(0,1+H_\beta(x))\}$, where $H_\beta(x)$ is a neural network without range restriction. Similarly as the equivalence between programs (1) and (18) and between (A2) and (A3), program (B14) is equivalent to

$$\begin{cases} \max_\beta K_{\text{hinge}}(\theta,\beta) & \text{with } \theta \text{ fixed}, \quad \text{(B15a)} \\ \min_\theta E_{p_*(x)}\left\{H_\beta(x)\right\} - E_{p_\theta(x)}\left\{H_\beta(x)\right\} & \text{with } \beta \text{ fixed}. \quad \text{(B15b)} \end{cases}$$

Although $H_\beta(x)$ is often referred to as a discriminator, we emphasize that $H_\beta(x)$ does not correspond to a discrimination probability or its logit, such as $D_\beta(x)$ or $h_\beta(x)$ in model (3). In fact, the following result can be directly deduced from the classification consistency of the hinge loss (Lin, 2002). The form of $H_{\beta_\theta^*}(x)$ below can also be found in the proof of Theorem 3.1 in Lim and Ye (2017).

**Proposition 5.** *For any fixed $\theta$, $K_{hinge}(\theta, \beta)$ is minimized by any $\beta_\theta^*$ such that $H_{\beta_\theta^*}(x) = sign(p_*(x) - p_\theta(x))$, where $sign(c) = -1$ or 1 if $c < 0$ or $c > 0$ and $sign(c) \in [-1, 1]$ if $c = 0$. With $\beta = \beta_\theta^*$, the objective function in (B15b) reduces to the total-variation distance*

$$TV(p_*, p_\theta) = \int |p_*(x) - p_\theta(x)| \, dx.$$

From this result, we see that if the "discriminator" $H_\beta(x)$ is sufficiently rich and well trained, then hinge GAN program (B14) leads to minimization of the total-variation distance $TV(p_*, p_\theta)$ not the reverse KL divergence as stated in Miyato et al. (2018). Therefore, hinge GAN is theoretically equivalent to the energy-based GAN (Zhao et al., 2017), which is shown in Arjovsky et al. (2017) to minimize the total-variation distance under an optimal discriminator.