THE DIVERGENCES MINIMIZED BY NON-SATURATING GAN TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpreting generative adversarial network (GAN) training as approximate divergence minimization has been theoretically insightful, has spurred discussion, and has lead to theoretically and practically interesting extensions such as f-GANs and Wasserstein GANs. For both classic GANs and f-GANs, there is an original variant of training and a "non-saturating" variant which uses an alternative form of generator gradient. The original variant is theoretically easier to study, but for GANs the alternative variant performs better in practice. The non-saturating scheme is often regarded as a simple modification to deal with optimization issues, but we show that in fact the non-saturating scheme for GANs is effectively optimizing a reverse KL-like f-divergence. We also develop a number of theoretical tools to help compare and classify f-divergences. We hope these results may help to clarify some of the theoretical discussion surrounding the divergence minimization view of GAN training.

1 INTRODUCTION

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have enjoyed remarkable progress in recent years, producing images of striking fidelity, resolution and coherence (Karras et al., 2018; Miyato et al., 2018; Brock et al., 2018; Karras et al., 2019). There has been much progress in both theoretical and practical aspects of understanding and performing GAN training (Nowozin et al., 2016; Arjovsky & Bottou, 2017; Arjovsky et al., 2017; Mescheder et al., 2018; Gulrajani et al., 2017; Sønderby et al., 2017; Miyato et al., 2018; Karras et al., 2018; Brock et al., 2018; Karras et al., 2019).

One of the key considerations for GAN training is the training scheme used to update the generator and critic. A rich avenue of developments has come from viewing GAN training as *divergence minimization*. This perspective dates back to Goodfellow et al. (2014), who showed that the original GAN training formulation can be viewed as approximately minimizing the Jensen-Shannon divergence. The f-GAN formulation (Nowozin et al., 2016) extended the range of divergences which could be minimized by GAN training to f-divergences such as reverse KL, using a principled approach based on a variational lower bound. Wasserstein GANs (Arjovsky et al., 2017), which approximately minimize the Wasserstein metric, have become quite popular due in part to their pleasing theoretical underpinning, ease of implementation, and strong practical results. Nevertheless a relatively unprincipled "non-saturating" scheme (Goodfellow et al., 2014) has continued to obtain groundbreaking results (Karras et al., 2019) and remains a state-of-the-art approach (Lucic et al., 2018).

The precise practical effect of the non-saturating scheme and whether it can be motivated in a principled way have been a source of discussion. The non-saturating scheme modifies the gradient used to update the generator. Goodfellow et al. (2014) motivates this modification as a simple trick to make gradients flow better, and claims it does not affect the "fixed point of the dynamics". We will see that this is not true in the parametric case. Poole et al. (2016) attempted to interpret the non-saturating scheme as approximate divergence minimization. However we will see that the proposed divergence does not match the gradient of the non-saturating loss, and so using this divergence for training does not replicate the non-saturating scheme. Arjovsky & Bottou (2017) derived a globally coherent objective function with the correct gradients which is approximately minimized by the non-saturating scheme. However they expressed the objective as a difference between two divergences, noting that

it is therefore a strange thing to minimize! In this paper we show that the non-saturating scheme approximately minimizes the f-divergence $4\,\mathrm{KL}(\frac{1}{2}p + \frac{1}{2}q \,\|\, p)$, which refer to as the *softened reverse KL divergence*. We also show that the non-saturating version of KL minimization is in fact reverse KL minimization.

In order to better understand the qualitative behavior of different divergences such as softened reverse KL, we develop several tools. While f-divergences unify many divergences, just plotting the function f is often not informative. The symmetric relationship between divergences such as KL and reverse KL are obfuscated, and f may grow quickly even when the divergence is well-behaved. We show how to write f-divergences in a symmetry-preserving way, allowing easy visual comparison of f-divergences in a way that reflects their qualitative properties. The most important f-divergence properties are in fact determined by just two numbers. We develop a rigorous formulation of *tail weight* which generalizes the notions of *mode-seeking* and *covering* behavior. Using these tools we show that the softened reverse KL divergence is fairly similar to the reverse KL but very different to the Jensen-Shannon divergence approximately minimized by the original GAN training scheme.

The remainder of the paper is structured as follows. We review the definition of f-divergences in §2. In §3 and §4 we develop the main tools we will use to compare f-divergences, including push-forwards, symmetry-preserving representations, tails weights and boundedness. In §5 we discuss operations on divergences. The softened reverse KL is described in terms of these operations. We recap the f-divergence approach to variational divergence estimation and minimization, also known as GAN training, in §6 and §7. We describe the non-saturating scheme in §8. In §9 we derive our main result, showing the divergences effectively minimized by various forms of non-saturating scheme. We discuss related previous work in §10. Finally we perform a basic experimental validation of our main mathematical result in §11.

2 THE FAMILY OF F-DIVERGENCES

In this section we review the definition of an f-divergence (Ali & Silvey, 1966), introduce terminology, and establish a number of relevant mathematical properties related to linearity, symmetry, limiting behavior for nearby distributions, boundedness and tail weight.

Given a strictly convex twice continuously differentiable function $f : \mathbb{R}_{>0} \to \mathbb{R}$, the f-divergence between probability distributions with densities[1] p and q over $\mathbb{R}^K$ is defined as:[2]

$$D_f(p, q) = \int q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \tag{1}$$

Typically p is the "true" or data distribution and q is the distribution of a model which is intended to approximate p.

We briefly summarize a few simple mathematical properties. Firstly note that D_f is linear f, that is $D_{f+g} = D_f + D_g$ and $D_{kf} = kD_f$ where $k > 0$. Secondly note that adding a constant or linear term to f only affects the D_f up to an overall additive constant: If $g(u) = f(u) + k + lu$ for $k, l \in \mathbb{R}$ then $D_g(p, q) = D_f(p, q) + k + l$. Thus the second derivative f'' determines the divergence up to an additive constant, and determines the gradients of the divergence completely. This property is also true of the various bounds and finite sample approximations[3] derived below, so we may legitimately consider f'' rather than f as the essential quantity of interest for a given divergence. For many common f-divergences, f'' has a simpler algebraic form than f. For any densities p and q we have $D_f(p, q) \geq f(1)$ with equality iff $p = q$, as can be seen by plugging the constant function $u(x) = 1$ into (12) below. This justifies referring to D_f as a divergence. If $f(1) = 0$, $f'(1) = 0$ and $f''(1) = 1$ then we say f is in *canonical form*. We can put any f in canonical form by scaling and adding a suitable constant-plus-linear term, and this corresponds to a simple shift and scale of D_f. Each f-divergence has a unique canonical form.

[1] Most results also hold for "discrete" probability distributions. The only difference is that the reparameterization trick can no longer be used to reduce variance of the finite sample approximations.

[2] For simplicity, we assume the probability distributions are suitably nice, e.g. absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^K$, $p(x), q(x) > 0$ for $x \in \mathbb{R}^K$, and p and q continuously differentiable.

[3] As long as the reparameterization trick (34) is used, as is standard practice. If a simpler finite sample approximation such as naive REINFORCE is used then k affects the variance of the generator gradient.

Different f-divergences may behave very differently when p and q are far apart, but are all closely related in the region where $q \approx p$. Specifically

$$D_f\left(q_\lambda, q_{\lambda+\varepsilon v}\right) = \tfrac{1}{2}\varepsilon^2 f''(1) v^\mathsf{T} F(\lambda) v + O(\varepsilon^3) \tag{2}$$

where $\varepsilon \in \mathbb{R}$, $v \in \mathbb{R}^K$, and $F(\lambda) = \sum_x q_\lambda(x)(\frac{\partial}{\partial\lambda}\log q_\lambda(x))(\frac{\partial}{\partial\lambda}\log q_\lambda(x))^\mathsf{T}$ is the Fisher information matrix for the parametric family of distributions specified by q_λ. Thus all f-divergences agree up to a constant factor on the divergence between two nearby distributions, and they are all just scaled versions of the Fisher metric in this regime. This can also be seen in Figure 2 below, where all f-divergences approximately overlap near zero.

The definition (1) appears to be quite asymmetric in how it treats p and q, but it obeys a particular symmetry (Reid & Williamson, 2011). It is straightforward to verify that if $f_\mathsf{R}(u) = uf(u^{-1})$ then $D_{f_\mathsf{R}}(p,q) = D_f(q,p)$ for any densities p and q. Differentiating twice, if $f''_\mathsf{R}(u) = u^{-3}f''(u^{-1})$ then $D_{f_\mathsf{R}}(p,q) = D_f(q,p) + k$ for some unimportant constant $k \in \mathbb{R}$. The converse is also true: If $D_{f_\mathsf{R}}(p,q) = D_f(q,p) + k$ for all densities p and q then $f''_\mathsf{R}(u) = u^{-3}f''(u^{-1})$. This can be seen for example by partitioning $\mathbb{R}^K$ into two sets A and B and considering densities p and q which are constant on A and constant on B, or strictly speaking smooth approximations thereof. Thus swapping the role of p and q corresponds to a particular transform of f''. We say D_f is *symmetric* if $D_f(p,q) = D_f(q,p)$ for all densities p and q. From the above we see that D_f is symmetric iff $f''(u) = u^{-3}f''(u^{-1})$. The above discussion allows us to rewrite (1) to be more explicitly symmetric in the role of p and q. With $A = \{x : q(x) > p(x)\}$ and $B = \{x : q(x) < p(x)\}$, we have

$$D_f(p,q) = \int_A q(x) f\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x + \int_B p(x)\frac{f\left(\frac{p(x)}{q(x)}\right)}{\frac{p(x)}{q(x)}}\,\mathrm{d}x \tag{3}$$

$$= \int_A q(x) f\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x + \int_B p(x) f_\mathsf{R}\left(\frac{q(x)}{p(x)}\right)\mathrm{d}x \tag{4}$$

which has clearer symmetry than (1). We refer to A as the set of *left mismatches* and B and the set of *right mismatches*. At each point in A, the two distributions p and q are somewhat mismatched, and the penalty paid for this mismatch in terms of the overall divergence D_f is governed by the behavior of $f(u)$ for $0 < u < 1$ (the "left" of the graph of f). Similarly the penalty paid for right mismatches is governed by $f(u)$ for $u > 1$. Note from (4) that a left mismatch can only be heavily penalized if the point is plausible under q, i.e. $q(x)$ is not tiny. Similarly a right mismatch can only be heavily penalized for points which are plausible under p.

3 Pushforwards and symmetry-preserving divergence plots

The considerations of symmetry in §2 lead to a straightforward and intuitive way to compare f-divergences visually, through a *symmetry-preserving divergence plot*. This perspective also allows a simple summary of the prevalence of mismatches between p and q, through a *pushforward plot*. In this section we develop this viewpoint and look at some examples of these plots.

Firstly note that for $x \sim q(x)$, $p(x)/q(x)$ is a random variable with some distribution. In fact, since any f-divergence (1) is the expected value of some function of this random variable, its value must depend only on the one-dimensional distribution of this random variable and not on the detailed distribution of p and q in space. Formally the distribution of this random variable may be described as the *pushforward measure* of q through the function $u^*(x) = p(x)/q(x)$. To obtain more intuitive plots, we will work in terms of $d^*(x) = \log p(x) - \log q(x)$ instead of u^*. We denote the density of the pushforward of q through d^* by $\tilde{q}_{d^*}(d)$. The expected value defining the f-divergence can thus be written

$$D_f(p,q) = \int \tilde{q}_{d^*}(d) f(\exp d)\,\mathrm{d}d \tag{5}$$

As above we can write this more symmetrically. Define

$$s_f(d) = \begin{cases} f(\exp d), & d < 0 \\ f_\mathsf{R}\left(\exp(-d)\right), & d > 0 \end{cases} \tag{6}$$

By considering expectations of an arbitrary function of d expressed in x-space and d-space, we can derive that

$$\tilde{q}_{d^*}(d) = \tilde{p}_{d^*}(d) \exp(-d) \tag{7}$$

Thus, using (4), we can write the f-divergence as

$$D_f(p, q) = \int_A q(x) s_f(d^*(x)) \, \mathrm{d}x + \int_B p(x) s_f(d^*(x)) \, \mathrm{d}x \tag{8}$$

$$= \int_{-\infty}^{0} \tilde{q}_{d^*}(d) s_f(d) \, \mathrm{d}d + \int_0^{\infty} \tilde{p}_{d^*}(d) s_f(d) \, \mathrm{d}d \tag{9}$$

or even more concisely, using (7), as

$$D_f(p, q) = \int_{-\infty}^{\infty} \max\left\{\tilde{p}_{d^*}(d), \tilde{q}_{d^*}(d)\right\} s_f(d) \, \mathrm{d}d \tag{10}$$

An f-divergence $D_f(p, q)$ involves an interaction between the distributions p, q and the function f, and (10) nicely decomposes this interaction in terms of something that only depends on p and q (the pushforwards) and something that only depends on f (the function s_f), connected via a one-dimensional integral. By plotting s_f and imagining integrating against various pushforwards, we can see the properties of different f-divergences in a very direct way. By plotting the pushforwards, we can get a feel for what types of mismatch between p and q are present in multidimensional space, and understand at a glance how badly these mismatches would be penalized for a given f-divergence.

Examples of pushforwards for the simple case where p and q are multidimensional Gaussians with common covariance are shown in Figure 1. In this case the pushforwards $\tilde{q}_{d^*}$ and $\tilde{p}_{d^*}$ are themselves one-dimensional Gaussians (since d^* is linear), with densities $\mathcal{N}(-\frac{1}{2}\sigma^2, \sigma^2)$ and $\mathcal{N}(\frac{1}{2}\sigma^2, \sigma^2)$ respectively, for some σ (this follows from (7)). For more complicated models, pushforward plots are straightforward to estimate empirically by using a learned $d(x)$ (see §6) instead of the optimal $d^*(x)$. This may be a very generally useful approach to monitoring the progression of GAN training. However we leave its investigation for future work. Examples of s_f for various f-divergences are shown in Figure 2. We refer to s_f as a *symmetry-preserving* representation of f. Note that as long as f is in canonical form, s_f is twice continuously differentiable at zero. Figure 2 directly expresses several facts about divergences. It shows that left mismatches (regions of space where $q(x) > p(x)$, corresponding to $d < 0$) are penalized by reverse KL much more severely than right mismatches (regions of space where $q(x) > p(x)$, corresponding to $d > 0$). The symmetry between KL and reverse KL is evident: a given left mismatch is penalized by KL the same amount a right mismatch of the same magnitude is penalized by reverse KL. We see that Jensen-Shannon and the Jeffreys divergence (the average of KL and reverse KL) are both symmetric in how they penalize left and right mismatches, but differ greatly in how much they penalize small versus large mismatches.

4 Classification of f-divergence tails

In this section we introduce a classification scheme for f-divergences in terms of their behavior for large left and right mismatches. While different f-divergences differ in details, this classification determines many aspects of their qualitative behavior.

First we define the notion of tail weight and examine some of its consequences. If $f''(u) \sim Cu^{-R}$ as $u \to 0$ for $C > 0$ and $f''(u) \sim Du^{S-3}$ as $u \to \infty$ for $D > 0$ then we say that D_f has (Cu^{-R}, Du^{S-3}) tails and (R, S) tail weights. Here we have used the notation $g(u) \sim h(u)$ as $u \to a$ to mean $g(u)/h(u) \to 1$ as $u \to a$. Note that, since $f''_{\mathrm{R}}(u) = u^{-3} f''(u^{-1})$, f having a u^{S-3} right tail is equivalent to f_{R} having a u^{-S} left tail. Thus tail weights interact simply with symmetry: If D_f has (R, S) tail weights then $D_{f_{\mathrm{R}}}$ has (S, R) tail weights, and if D_f is symmetric then its left and right tail weights are equal. Tail weights also interact in a simple and intuitive way with linearity: If one f-divergence has (R_1, S_1) tail weights and another has (R_2, S_2) tail weights then their sum has $(\max_i R_i, \max_i S_i)$ tail weights. Intuitively, the left tail weight R determines how strongly large left mismatches are penalized, whereas the right tail weight S determines how strongly large right mismatches are penalized.

Some f-divergences such as Jensen-Shannon are bounded, while others such as KL are unbounded, and it will be useful to have a characterization of when boundedness occurs. We say D_f is bounded

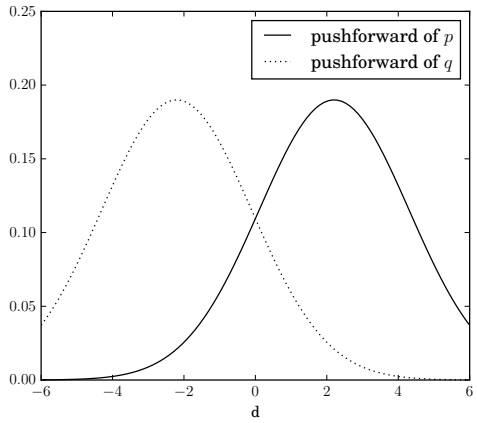

Figure 1: Plots of the pushforward densities $\tilde{p}_{d^*}(d)$ and $\tilde{q}_{d^*}(d)$ for the case where p and q are multidimensional Gaussians with common covariance. The f-divergence for a given f may be obtained by integrating these pushforwards against s_f in Figure 2 using (10).

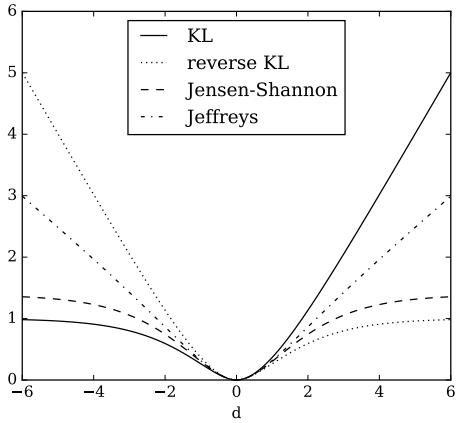

Figure 2: Plots of $s_f(d)$ for various f-divergences. The f-divergence for a given p and q may be obtained by integrating s_f against the pushforwards of p and q such as those shown in Figure 1 using (10). Symmetries such as that between KL and reverse KL are evident.

divergence	tail weights	(left, right) boundedness	overall boundedness
KL	$(1, 2)$	$(0, \infty)$	∞
reverse KL	$(2, 1)$	$(\infty, 0)$	∞
Jensen-Shannon	$(1, 1)$	$(0, 0)$	0
Jeffreys ($\frac{1}{2}$KL + $\frac{1}{2}$RKL)	$(2, 2)$	(∞, ∞)	∞
Pearson χ^2	$(3, 0)$	$(\infty, 0)$	∞
softened KL	$(1, 0)$	$(0, 0)$	0
softened reverse KL (SRKL)	$(2, 0)$	$(\infty, 0)$	∞

Table 1: Tail weights and boundedness for the f-divergences considered in this paper. For tail weights, the notation (R, S) denotes a left tail weight of R and a right tail weight of S. A divergence is left-bounded iff $R < 2$, right-bounded iff $S < 2$, and bounded overall iff both $R < 2$ and $S < 2$. For boundedness, 0 denotes bounded and ∞ denotes unbounded, so for example $(0, \infty)$ denotes left-bounded and right-unbounded.

if there is an $M \in \mathbb{R}$ such that $D_f(p, q) \leq M$ for all densities p and q. We say f is *left-bounded* if f is bounded on $(0, 1)$, and *right-bounded* if f_{R} is bounded on $(0, 1)$, or equivalently if $f(u)/u$ is bounded on $u > 1$. From (4) it is easy to see that if f is left-bounded and right-bounded then D_f is bounded. The converse is also true: If f is left-unbounded or right-unbounded then we can find p and q with arbitrarily large divergence $D_f(p, q)$. This can be seen for example by partitioning $\mathbb{R}^K$ into two sets A and B and considering densities p and q which are constant on A and constant on B, or strictly speaking smooth approximations thereof.

Tail weight determines boundedness. It can be checked by integrating and bounding that a divergence with (R, S) tail weights is left-bounded iff $R < 2$ and right-bounded iff $S < 2$. Thus D_f is bounded iff $R, S < 2$. The tail weights and boundedness properties of various f-divergences considered in this paper are summarized in Table 1. Boundedness properties can also be seen in Figure 2. Left and right boundedness of f is trivially equivalent to left and right boundedness of s_f. Thus we can see that reverse KL is left unbounded but right bounded, for example. The unbounded tails in this plot are all asymptotically linear in d.

Tail weights provide an extension of the typical classification of divergences as *mode-seeking* or *covering*. Models trained with reverse KL tend to have distributions which are more compact than

the true distribution, sometimes only successfully modeling certain modes (density peaks) of a multimodal true distribution. Models trained with KL tend to have distributions which are less compact than the true distribution, "covering" the true distribution entirely even if it means putting density in regions which are very unlikely under the true distribution. However there are important qualitative aspects of divergence behavior that are not captured by these labels. For example, is Jensen-Shannon mode-seeking or covering? Really, it is neither: It would be more accurate to say that a model trained Jensen-Shannon tries to match very closely when it matches, but doesn't worry overly about large mismatches in either direction. The Jeffreys divergence is also symmetric and so neither mode-seeking nor covering, but has very different behavior from Jensen-Shannon. Tail weights capture these distinctions in a straightforward but precise way.

5 DIVERGENCE SYMMETRIZATION AND SOFTENING

We can apply some simple operations to a divergence to obtain another divergence. In this section we consider the effect of reversing, symmetrizing and *softening* operations on f-divergences. Many common f-divergences can be obtained from others in this way, and this provides a unified way of concisely describing f-divergences based on KL.

Consider applying an operation to a divergence $D(p, q)$ to obtain another divergence $\tilde{D}(p, q)$. We have already seen the *reversing* operation $\tilde{D}(p, q) = D(q, p)$. If D is an f-divergence with function $f(u)$ then D_{R} is an f-divergence with function $f_{\mathrm{R}}(u) = uf(u^{-1})$. In this case $f''_{\mathrm{R}}(u) = u^{-3}f''(u^{-1})$. Symmetrization means $\tilde{D}(p, q) = \frac{1}{2}D(p, q) + \frac{1}{2}D(q, p)$. If D is an f-divergence then $f \mapsto \frac{1}{2}f + \frac{1}{2}f_{\mathrm{R}}$ enacts symmetrization. Finally *(q-)softening* refers to replacing q with $m = \frac{1}{2}p + \frac{1}{2}q$, i.e. $\tilde{D}(p, q) = 4D(p, m)$. If D is an f-divergence with function f then setting the new $f(u)$ to be $4\frac{1+u}{2}f(\frac{2u}{1+u})$ enacts softening. In this case the new $f''(u)$ is $\frac{8}{(1+u)^3}f''(\frac{2u}{1+u})$. The factor of 4 above is to ensure that the divergence remains canonical after softening, i.e. $f''(1) = 1$. Softening has the potential to make large right mismatches much less severely penalized, since in regions of space where $p(x)/q(x)$ was large because $p(x)$ was moderate and $q(x)$ was tiny, $p(x)/m(x)$ is now approximately 2, so a large right mismatch is only penalized by the softened divergence as much as a moderate right mismatch is penalized by the original divergence. This is reflected in the tail weights: It is easy to show using the tools we have developed above that if the original divergence has (R, S) tail weights then the softened divergence has $(R, 0)$ tail weights.

Many f-divergences can be written concisely as a series of these operations. For example reverse KL is Reverse(KL), Jeffreys is Symmetrize(KL), the K-divergence $4\,\mathrm{KL}(p \,\|\, m)$ (Cha, 2007) is Soften(KL) and Jensen-Shannon is Symmetrize(Soften(KL)). In this terminology, the main claim of this paper is that the non-saturating procedure for GAN training is in fact effectively minimizing the softened reverse KL divergence $4\,\mathrm{KL}(m \,\|\, p)$ given by Soften(Reverse(KL)).

6 VARIATIONAL DIVERGENCE ESTIMATION

f-GANs are based on an elegant way to estimate the f-divergence between two distributions given only samples from the two distributions (Nguyen et al., 2010). In this section we review this approach to *variational divergence estimation*.

There is an elegant variational bound on the f-divergence $D_f(p, q)$ between two densities p and q. Since f is strictly convex, its graph lies at or above any of its tangent lines and only touches in one place. That is, for $k, u > 0$,

$$f(k) \geq f(u) + (k - u)f'(u) = kf'(u) - \big[uf'(u) - f(u)\big] \tag{11}$$

with equality iff $k = u$. This inequality is illustrated in Figure 3. Substituting $p(x)/q(x)$ for k and $u(x)$ for u, for any continuously differentiable function $u : \mathbb{R}^K \to \mathbb{R}_{>0}$ we obtain

$$D_f(p, q) \geq \int p(x)f'(u(x))\,\mathrm{d}x - \int q(x)\left[u(x)f'(u(x)) - f(u(x))\right]\mathrm{d}x \tag{12}$$

with equality iff $u = u^*$, where $u^*(x) = p(x)/q(x)$. The function u is referred to as the *critic*. It will be helpful to have a concise notation for this bound. Writing $u(x) = \exp(d(x))$ without loss of

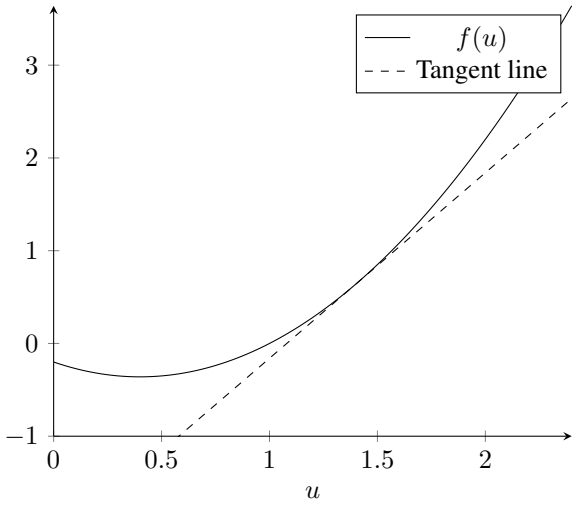

Figure 3: A strictly convex function $f : \mathbb{R}_{>0} \to \mathbb{R}$ and a tangent line. The variational bound used by f-GANs is based on the fact that a strictly convex function f lies at or above its tangent lines.

generality, for any continuously differentiable function $d : \mathbb{R}^K \to \mathbb{R}$, we have

$$D_f(p, q) \geq E_f(p, q, d) \tag{13}$$

with equality iff $d = d^*$, where

$$E_f(p, q, d) = \int p(x) a_f(d(x)) \, \mathrm{d}x - \int q(x) b_f(d(x)) \, \mathrm{d}x \tag{14}$$

$$a_f(d) = f'(\exp(d)) \tag{15}$$

$$b_f(d) = \exp(d) f'(\exp(d)) - f(\exp(d)) \tag{16}$$

$$d^*(x) = \log p(x) - \log q(x) \tag{17}$$

Note that both a_f and b_f are linear in f. Their derivatives

$$a'_f(\log u) = u f''(u) \tag{18}$$

$$b'_f(\log u) = u^2 f''(u) \tag{19}$$

depend on f only through f''.

The above formulation naturally leads to *variational divergence estimation*. The f-divergence between p and q can be estimated by maximizing E_f with respect to d (Nguyen et al., 2010). Conveniently E_f is expressed in terms of expectations and may be approximately computed and maximized with respect to d using only samples from p and q. If we parameterize d as a neural net d_ν with parameters ν then we can approximate the divergence by maximizing $E_f(p, q, d_\nu)$ with respect to ν. This does not compute the exact divergence because there is no guarantee that the optimal function d^* lies in the family $\{d_\nu : \nu\}$ of functions representable by the neural net, but we hope that for sufficiently flexible neural nets the approximation will be close.

The original f-GAN paper (Nowozin et al., 2016) phrases the above results in terms of the Legendre transform f^* of f. The two descriptions are equivalent, as can be seen by setting $T(x) = f'(u(x))$ and using the result $f^*(f'(u)) = u f'(u) - f(u)$. We find our description helpful since it avoids having to explicitly match the domain of f^*, ensures the optimal d is the same for all f-divergences, and because the Legendre transform is complicated for one of the divergences we consider. An "output activation" was used in the original f-GAN paper to adapt the output d of the neural net to the domain of f^*. This is equal to $f'(\exp(d))$, up to irrelevant additive constants, for all the divergences we consider, and so our description also matches the original description in this respect.

Here we briefly summarize the three main f-divergences we consider. The expressions for D_f and E_f are obtained by plugging the chosen f into (1) and (14) respectively. The *Kullback-Leibler (KL)*

divergence satisfies:

$$f(u) = u \log u \tag{20}$$
$$f''(u) = u^{-1} \tag{21}$$
$$D_f(p, q) = \mathrm{KL}(p \,\|\, q) = \int p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x \tag{22}$$
$$E_f(p, q, d) = 1 + \int p(x) d(x) \,\mathrm{d}x - \int q(x) \exp(d(x)) \,\mathrm{d}x \tag{23}$$

Using the notation established in §2, the KL divergence has (u^{-1}, u^{-1}) tails, $(1, 2)$ tail weights, and is left-bounded and right-unbounded.

The *reverse KL divergence* satisfies:

$$f(u) = -\log u \tag{24}$$
$$f''(u) = u^{-2} \tag{25}$$
$$D_f(p, q) = \mathrm{KL}(q \,\|\, p) = \int q(x) \log \frac{q(x)}{p(x)} \,\mathrm{d}x \tag{26}$$
$$E_f(p, q, d) = 1 - \int p(x) \exp(-d(x)) \,\mathrm{d}x - \int q(x) d(x) \,\mathrm{d}x \tag{27}$$

The reverse KL divergence has (u^{-2}, u^{-2}) tails, $(2, 1)$ tail weights, and is left-unbounded and right-bounded.

If we make the *Jensen Shannon (JS) divergence* canonical by multiplying it by 4, it is defined as

$$f(u) = 2u \log u - 2(u + 1) \log(u + 1) + 4 \log 2 \tag{28}$$
$$f''(u) = \frac{2}{u(u + 1)} \tag{29}$$
$$D_f(p, q) = 4 \,\mathrm{JS}(p, q) \tag{30}$$
$$= 2 \,\mathrm{KL}(p \,\|\, \tfrac{1}{2}p + \tfrac{1}{2}q) + 2 \,\mathrm{KL}(q \,\|\, \tfrac{1}{2}p + \tfrac{1}{2}q) \tag{31}$$
$$= 4 \log 2 + 2 \int p(x) \log \frac{p(x)}{p(x) + q(x)} \,\mathrm{d}x + 2 \int q(x) \log \frac{q(x)}{p(x) + q(x)} \,\mathrm{d}x \tag{32}$$
$$E_f(p, q, d) = 4 \log 2 + 2 \int p(x) \log \sigma(d(x)) \,\mathrm{d}x + 2 \int q(x) \log \sigma(-d(x)) \,\mathrm{d}x \tag{33}$$

where JS denotes the conventional definition of the Jensen-Shannon divergence. The 4 JS divergence has $(2u^{-1}, 2u^{-2})$ tails, $(1, 1)$ tail weights, and is both left-bounded and right-bounded and so bounded overall.

7 VARIATIONAL DIVERGENCE MINIMIZATION

f-GANs (Nowozin et al., 2016) generalize classic GANs (Goodfellow et al., 2014) to allow approximately minimizing any f-divergence. The Jensen-Shannon divergence optimized by GANs is an f-divergence. In this section we briefly review and discuss the f-GAN formulation (Nowozin et al., 2016).

Consider the task of estimating a probabilistic model from data using an f-divergence. Here p is the true distribution and the goal is to minimize $l(\lambda) = D_f(p, q_\lambda)$ with respect to λ, where $\lambda \mapsto q_\lambda$ is a parametric family of densities over $\mathbb{R}^K$. We refer to q_λ as the *generator*. For implicit generative models such as GANs, q_λ is defined implicitly: x is assumed to be the result $\overline{x}_\lambda(z)$ of transforming a stochastic latent variable z with fixed distribution by a parameterized deterministic neural network $\overline{x}_\lambda$.

$$E_f(p, q_\lambda, d) \stackrel{\mathrm{c}}{=} \int q_\lambda(x) \log \sigma\left(-d(x)\right) \mathrm{d}x = \int \mathbb{P}(z) \log \sigma\left(-d(\overline{x}_\lambda(z))\right) \mathrm{d}z \tag{34}$$

However we do not need to assume this specific form for most of our discussion.

We first note that the variational divergence bound E_f satisfies a convenient gradient matching property. This is not made explicit in the original f-GAN paper. Denote the optimal d given p and q_λ by d^*_λ. We saw above that $D_f(p, q_\lambda)$ and $E_f(p, q_\lambda, d)$ match values at $d = d^*_\lambda$. They also match gradients:

$$\frac{\partial}{\partial \lambda} D_f(p, q_\lambda) = \frac{\partial}{\partial \lambda} E_f(p, q_\lambda, d) \bigg|_{d=d^*_\lambda} = -\int \left[\frac{\partial}{\partial \lambda} q_\lambda(x)\right] b_f(d^*_\lambda(x))\, \mathrm{d}x \tag{35}$$

This follows from the fact that E_f is a tight lower bound on D_f, similarly to the one-dimensional result that any differentiable function $f : \mathbb{R} \to \mathbb{R}$ with $f(x) \geq 0$ for all x and $f(0) = 0$ has $f'(0) = 0$. It is also straightforward to verify (35) directly from the definitions of D_f and E_f.

We can minimize $D_f(p, q_\lambda)$ using *variational divergence minimization*, maximizing $E_f(p, q_\lambda, d_\nu)$ with respect to ν while minimizing it with respect to λ. Adversarial optimization such as this lies at the heart of all flavors of GAN training. Define $\overline{\lambda}$ and $\overline{\nu}$ as

$$\overline{\lambda} = -\frac{\partial}{\partial \lambda} E_f(p, q_\lambda, d_\nu) = \int \left[\frac{\partial}{\partial \lambda} q_\lambda(x)\right] b_f(d_\nu(x))\, \mathrm{d}x \tag{36}$$

$$\overline{\nu} = \frac{\partial}{\partial \nu} E_f(p, q_\lambda, d_\nu) \tag{37}$$

To perform the adversarial optimization, we can feed $\overline{\lambda}$ and $\overline{\nu}$ (or in practice, stochastic approximation to them) as the gradients into any gradient-based optimizer designed for minimization, e.g. stochastic gradient descent or ADAM. The gradient matching property shows that performing very many critic updates followed by a single generator update is a sensible learning strategy which, assuming the critic is sufficiently flexible and amenable to optimization, essentially performs very slow gradient-based optimization on the true divergence D_f with respect to λ. However in practice performing a few critic updates for each generator update, or simultaneous generator and critic updates, performs well, and it is easy to see that these approaches at least have the correct fixed points in terms of Nash equilibria of E_f and optima of D_f, subject as always to the assumption that the critic is sufficiently richly parameterized. Convergence properties of these schemes are investigated much more thoroughly elsewhere, for example (nag; Gulrajani et al., 2017; Mescheder et al., 2017; 2018; Balduzzi et al., 2018; Peng et al., 2019), and are not the main focus here.

There is a simple generalization of the above training procedure, which is to base the generator gradients on E_f but the critic gradients on E_g for a possibly different function g (Poole et al., 2016, section 2.2). Subject as always to the assumption of a richly parameterized critic, if we perform very many critic updates for each generator update, then the d used to compute the generator gradient will still be close to d^*, and so the generator gradient will be close to the gradient of D_f, even though the path d took to approach d^* was governed by g rather than f. The fixed points of the two gradients are also still correct, and so it seems reasonable to again use more general update schemes and we might hope for similar convergence results (not analyzed here). We refer to using gradients based on f to optimize the generator and gradients based on g to optimize the critic as using *hybrid* (f, g) gradients. For example, hybrid (KL, reverse KL) denotes optimizing the KL divergence using a critic trained based on reverse KL. Hybrid schemes were described by Poole et al. (2016).

8 Practical issues with training

When training classic GANs in practice, an alternative *non-saturating* loss is used as the basis for the generator gradient, and is found to perform much better in practice (Goodfellow et al., 2014). An analogous alternative generator gradient was also found to be useful for f-GANs (Nowozin et al., 2016). In this section we review the saturation issue with the original loss and describe the alternative loss.

We first discuss the "saturation" issue that occurs when optimizing bounded f-divergences. Early on in training, the generator and data distribution are typically not well matched, with samples from p being very unlikely under q and vice versa. This means most of the probability mass of p and q is in regions where d has large magnitude, corresponding to the positive and negative tails in Figure 2 and (10). For an implicit generative model $x_\lambda(z)$ where $z \sim \mathbb{P}(z)$, we have

$$E_f(p, q_\lambda, d) \stackrel{\mathrm{c}}{=} -\int \mathbb{P}(z) b_f\left(d_\nu(\overline{x}_\lambda(z))\right) \mathrm{d}z \tag{38}$$

Thus there is a $b'_f(d)$ factor in the generator gradient, and in fact this is the only way the choice of f-divergence affects the generator gradient. For reverse KL, $b'_f(d) = 1$, allowing the gradients from the other factors to pass freely. Most of the contribution to the initial gradient for reverse KL is likely to come from regions in space with large negative d due to the $\mathbb{P}(z)$ factor. For (four times) Jensen-Shannon, $b'_f(d) = 2\sigma(d)$, which tends to zero exponentially quickly as $d \to -\infty$ and tends to 2 as $d \to \infty$. Regions of space with large positive d have a tiny contribution to the gradient due to the $\mathbb{P}(z)$ factor, while regions with large negative d are exponentially suppressed by $b'_f(d)$. Based on these considerations it might be tempting to conclude that left-unboundedness is the most important factor in being able to learn from a random initialization. A divergence with left tail weight R has $b'_f(d) \sim \exp(-d(R-2))$ so $R \geq 2$ ensures that $b'_f(d)$ does not decay exponentially as $d \to -\infty$. However the case of KL shows that right-unboundedness is also capable of allowing learning. For KL, $b'_f(d) = \exp d$, and the situation is complicated, since it exponentially magnifies gradients from regions with large positive d, which are extremely unlikely under $\mathbb{P}(z)$. We know the overall gradient can sometimes be a reasonable learning signal, since training models such as a multivariate Gaussian using KL divergence works well. However even if the expected gradient allows learning, the stochastic approximation obtained by sampling from q is likely to have extremely large variance.

The saturation issue is sometimes presented as being specific to the loss E_f used for classic GAN training, but the gradient matching property presented in §7 shows it is fundamental to the Jensen-Shannon divergence. The more critic updates we perform initially, the more saturated d is on samples from q, and the more closely the gradient of E_f with respect to λ approximates the gradient of the true divergence D_f.

The typical fix to the saturation issue is to use the alternative generator gradient

$$\overline{\lambda} = \int \left[\frac{\partial}{\partial \lambda} q_\lambda(x)\right] \log \sigma\left(d(x)\right) \mathrm{d}x = \int \mathbb{P}(z) \left[\frac{\partial}{\partial \lambda} \log \sigma\left(d(\overline{x}_\lambda(z))\right)\right] \mathrm{d}x \tag{39}$$

Since the gradient of $\log \sigma(d)$ tends to 1 as d tends to $-\infty$, the gradient used for training is now larger. We have seen that other f-divergences such as KL divergence and reverse KL divergence are not bounded and do not suffer from the same saturation issue. Nevertheless an analogous alternative generator gradient was suggested for use in practice for f-GANs (Nowozin et al., 2016). Instead of the generator gradient being

$$\overline{\lambda} = \int \left[\frac{\partial}{\partial \lambda} q_\lambda(x)\right] b_f(d(x)) \, \mathrm{d}x \tag{40}$$

it is now

$$\overline{\lambda} = \int \left[\frac{\partial}{\partial \lambda} q_\lambda(x)\right] a_f(d(x)) \, \mathrm{d}x \tag{41}$$

It is easy to verify that this gives the same generator gradient as (39) when f is the Jensen-Shannon divergence. We refer to (41) as the *alternative* or *non-saturating* generator gradient, even though for general f-GANs "non-saturating" is a bit of a misnomer. The critic gradient is still given by (37).

9 Effect of non-saturating gradients

A natural question is what effect using the alternative generator gradient has on the final learned model. Does training using the alternative generator gradient approximately minimize the same divergence as training with the original generator gradient? Indeed the non-saturating loss is often presented as a practical afterthought to the theoretical discussion and viewed as a simple tweak to aid optimization (Goodfellow et al., 2014; Nowozin et al., 2016). However in this section we show that training using the alternative generator gradient effectively optimizes a different divergence. We also derive this divergence explicitly for some common cases. Finally we investigate some properties of the divergence effectively optimized by the alternative generator gradient when the original divergence is Jensen-Shannon, which is the form of gradient most commonly used in practice to train GANs.

We first establish the main result of this note: "Non-saturating" training based on g is precisely equivalent to a hybrid (f, g) scheme for some f. Suppose

$$f''(u) = u^{-1} g''(u) \tag{42}$$

Then it is straightforward to verify that $b'_f = a'_g$, so $b_f = a_g + k$, where the constant $k \in \mathbb{R}$ is irrelevant to the critic and generator gradients as mentioned in §7. Thus from (40) and (41) we see that an original generator gradient using f is the same as an alternative generator gradient using g. Such an f can always be found for any g: By integrating $u^{-1}g''(u)$ twice, we can find an f with the desired second derivative, and it is strictly convex since $g''(u) > 0$ and so $f''(u) = u^{-1}g''(u) > 0$ for $u > 0$. The critic gradient is still based on g, and so the overall scheme is precisely a hybrid (f, g) one. As discussed in §7, a hybrid (f, g) scheme is designed to optimize D_f. In our view it is thus incorrect to think of "non-saturating" training as a way to optimize D_g as is sometimes suggested.

In the remainder of this section we explicitly compute the corresponding f for some common choices of g. It is easy to show that if g has (R, S) tails then the corresponding f has $(R+1, S-1)$ tails, so the divergence effectively optimized by non-saturating training penalizes left mismatches more strongly and right mismatches less strongly than the original divergence.

For the KL divergence $g(u) = u \log u$, we have $g''(u) = u^{-1}$, so we need $f''(u) = u^{-2}$. We already saw in §6 that this f corresponds to the reverse KL divergence $f(u) = -\log u$. The equivalence of the alternative KL and original reverse KL generator gradients may also be seen directly from (23) and (27) by noting that $a_g(d) = d$ for the KL divergence is equal to $b_f(d) = d$ for the reverse KL divergence (since a plays the same role in the alternative generator gradient (41) as b plays in the original generator gradient (40) as we saw in §8). Thus "non-saturating" training based on the KL divergence is in fact a hybrid (reverse KL, KL) scheme, and so in fact optimizes the reverse KL divergence.

For the reverse KL divergence $g(u) = -\log u$, we have $g''(u) = u^{-2}$, so we need $f''(u) = u^{-3}$. Integrating twice, using constants of integration judiciously chosen to give a nice expression for D_f, we obtain $f(u) = \frac{1}{2}u^{-1}(u-1)^2$. The corresponding divergence is half the *Pearson χ^2 (or Kagan) divergence*. It satisfies:

$$f(u) = \frac{(u-1)^2}{2u} \tag{43}$$

$$f''(u) = u^{-3} \tag{44}$$

$$D_f(p, q) = \tfrac{1}{2} \int \frac{\big(q(x) - p(x)\big)^2}{p(x)} \, \mathrm{d}x \tag{45}$$

$$E_f(p, q, d) = -\tfrac{1}{2} - \tfrac{1}{2} \int p(x) \exp\big(-2d(x)\big) \, \mathrm{d}x + \int q(x) \exp\big(-d(x)\big) \, \mathrm{d}x \tag{46}$$

This divergence has (u^{-3}, u^{-3}) tails, $(3, 0)$ tail weights, and is left-unbounded and right-bounded. Again the equivalence of the two generator gradients may also be seen directly from (27) and (46) by noting that $a_g(d) = -\exp(-d)$ for the reverse KL divergence is equal to $b_f(u) = -\exp(-d)$ for half the Pearson χ^2 divergence. The expression for f here corrects a swapped definition in the original f-GAN paper[4] (according to the definitions of the Pearson and Neyman divergences given in the paper, the expression given for the Pearson f is actually the Neyman f and vice versa) (Nowozin et al., 2016). Thus "non-saturating" training based on in fact a hybrid ($\frac{1}{2}\chi^2$, reverse KL) scheme, and so in fact optimizes the Pearson χ^2 divergence.

For Jensen-Shannon-times-4 divergence, $g''(u) = \frac{2}{u(u+1)}$, so we need $f''(u) = \frac{2}{u^2(u+1)}$. Integrating twice, we obtain $f(u) = 2(u+1)\log\frac{u+1}{u} - 4\log 2$. The corresponding divergence does not have a prior name as far as we are aware. In this paper we have termed it the *softened reverse KL*

[4] In the the arxiv preprint, not the final NIPS version of the paper.

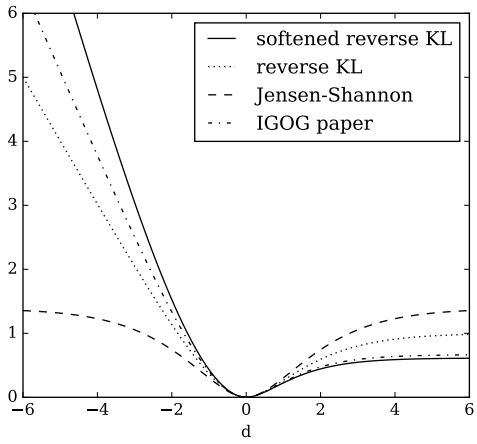

Figure 4: Plots of $s_f(d)$ for various reverse KL-like f-divergences. Softened reverse KL is the divergence effectively minimized by non-saturating GAN training. IGOG is the divergence derived by Poole et al. (2016).

(SRKL) divergence. It satisfies:

$$f(u) = 2(u+1)\log\frac{u+1}{u} - 4\log 2 \tag{47}$$

$$f''(u) = \frac{2}{u^2(u+1)} \tag{48}$$

$$D_f(p,q) = 4\,\mathrm{KL}(\tfrac{1}{2}p + \tfrac{1}{2}q \,\|\, p) \tag{49}$$

$$= -4\log 2 + 2\int \big(p(x)+q(x)\big)\log\frac{p(x)+q(x)}{q(x)}\,\mathrm{d}x \tag{50}$$

$$\begin{aligned} E_f(p,q,d) &= 2 - 4\log 2 + 2\int p(x)\left[-\exp\big(-d(x)\big) - \log\sigma\big(d(x)\big)\right]\mathrm{d}x \\ &\quad - 2\int q(x)\log\sigma\big(d(x)\big)\,\mathrm{d}x \end{aligned} \tag{51}$$

The SRKL divergence has $(2u^{-2}, 2u^{-3})$ tails, $(2, 0)$ tail weights, and is left-unbounded and right-bounded. Again the equivalence of the two generator gradients may also be seen directly from (33) and (51) by noting that $a_g(d) = 2\log\sigma(d)$ for the 4 JS divergence is equal to $b_f(d) = 2\log\sigma(d)$ for the SRKL divergence. Thus the conventional GAN non-saturating training scheme (Goodfellow et al., 2014) is in fact a hybrid (SRKL, JS) scheme, and so in fact optimizes the softened reverse KL divergence $4\,\mathrm{KL}(\frac{1}{2}p + \frac{1}{2}q \,\|\, p)$.

We can now use the tools developed in the first part of the paper to compare the qualitative effect of using the non-saturating variant of GAN training. Figure 4 shows the symmetry-preserving representation $s_f(d)$ for the Jensen-Shannon and softened reverse KL divergences, as well as the reverse KL for comparison. For reference the three divergences have tail weights (1, 1), (2, 0) and (2, 1) respectively. The qualitative behavior of softened reverse KL is quite similar to reverse KL. We noted in §5 that softening has the potential to make large right mismatches much less severely penalized, thus making the divergence more mode-seeking. However the reverse KL already penalizes right mismatches lightly and is mode-seeking. Softening decreases the tail weight from 1 to 0 and increases the slope of the left tail, but these changes are relatively minor modifications. They are both left-unbounded, right-bounded divergences. The Jensen-Shannon is extremely different to the reverse KL and softened reverse KL.

10 PREVIOUS DISCUSSION OF NON-SATURATING GRADIENTS

In this section we review some of this previous discussion in the context of the results presented here.

The original GAN paper claims: "This objective function results in the same fixed point of the dynamics of G and D but provides much stronger gradients early in learning." (Goodfellow et al., 2014, section 3). Note that it is true that the original and alternative generator gradients give the same final result in the non-parametric case where q is unrestricted, but this is fairly trivial since both gradients lead to $q = p$. It is even true that the dynamics are essentially the same for the original and alternative gradients when $q \approx p$, since the alternative gradient-based optimization minimizes a different f-divergence and we saw above that all f-divergences agree up to a constant in this regime, but again this is somewhat trivial since there is no sense in which the alternative Jensen-Shannon gradient is any more similar to the original Jensen-Shannon gradient than it is to any other original or alternative f-divergence gradient. The "fixed point of the dynamics" is certainly not the same in the general case of parametric q.

The original f-GAN paper presents a simple argument (which is apparently adapted from an argument in the original GAN paper (Goodfellow et al., 2014), though we could not find this) that the "non-saturating" training scheme has the same fixed points (Nowozin et al., 2016, section 3.2)[5]. However this argument is erroneous. It is true that if $p \approx q$ then $(f^*)'(f'(u))$ is approximately 1 everywhere, and so the original and alternative generator gradients are approximately equal. However there is no guarantee that the regime $p \approx q$ will ever be approached in the general parametric case, and as we will see it is not the case that the original and alternative generator gradients point in approximately the same direction in general.

A recent paper showed experimentally that the non-saturating generator gradient can successfully learn a distribution in a case where optimizing Jensen-Shannon divergence should fail, and used this to argue that perhaps it is not particularly helpful to view GANs as optimizing Jensen-Shannon divergence (Fedus et al., 2018). The divergence optimized in practice for parametric critics is not exactly the desired divergence, so it is conceivable that something might work experimentally that should not when viewed in an idealized way. Indeed in the situation where p and q initially have non-overlapping support, all the divergences we consider here are either ∞ or $\log 4$, so there is no gradient. However in this case we would argue the success in practice is probably as much due to optimizing a different divergence which has reasonable initial gradients as it is due to an inexact critic.

Arjovsky and Bottou correctly recognize that the alternative generator gradient results in optimizing a different divergence-like function and derive the function for classic GANs (Arjovsky & Bottou, 2017, section 2.2.2). The divergence-like function there is expressed as

$$\mathrm{KL}(q \,\|\, p) - 2\,\mathrm{JS}(p, q) \tag{52}$$

which is a slightly convoluted form of the expression $2\,\mathrm{KL}(\frac{1}{2}p + \frac{1}{2}q \,\|\, p)$ given in (49). The paper suggests the negative sign of the second term is "pushing for the distributions to be different, which seems like a fault in the update", whereas our expression for the divergence makes it clear that this is not an issue.

Poole et al. (2016) present a very similar view to that presented in this paper, including recognizing that the generator and critic may be trained to optimize different f-divergences and interpreting the classic non-saturating generator gradient as a hybrid scheme of this form where the generator gradient is based on a new f-divergence (Poole et al., 2016). However the f-divergence derived there is $f(u) = \log(1+u^{-1})$, which differs from (47) by a factor of $u+1$. We refer to this as the *improved generator objectives for GANs (IGOG)* divergence. It has $f''(u) = u^{-2} - (1+u)^{-2} = \frac{2u+1}{(1+u)^2u^2}$. Figure 4 shows that this divergence is qualitatively quite similar to the softened reverse KL but is not identical. The IGOG divergence has $(2, 0)$ tail weights. We now discuss the discrepancy. In the language of this paper, they use the approximation:

$$D_f(p, q) = \int q(x) f(p(x)/q(x))\,\mathrm{d}x \approx \tilde{E}(p, q, d) = \int q(x) f(\exp(d(x)))\,\mathrm{d}x \tag{53}$$

[5]Only in the final NIPS version of the paper, not the arxiv preprint.

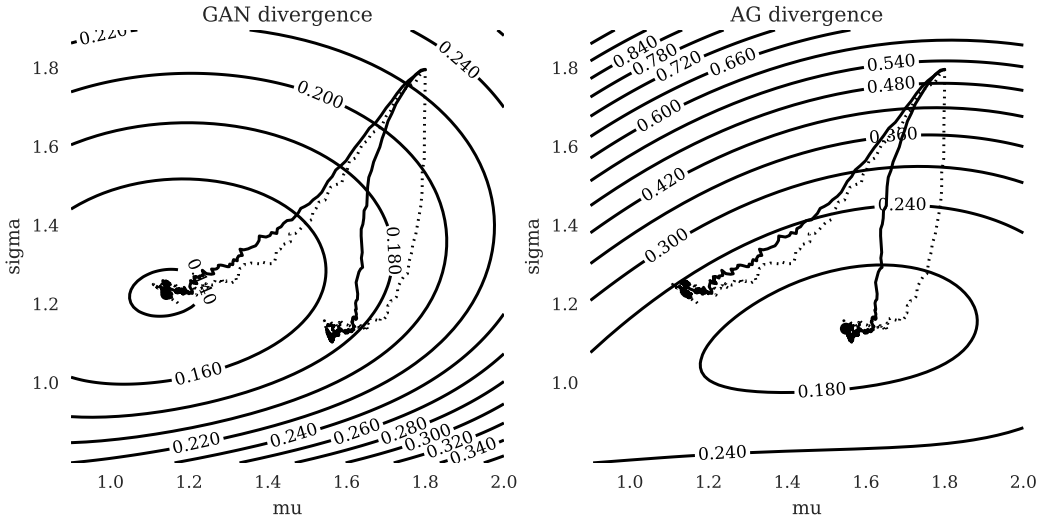

Figure 5: Comparing training using the saturating and non-saturating GAN generator gradients on a toy problem. The true distribution p is a mixture of two 1D Gaussians and the model distribution q is a single Gaussian. Contour plots show the Jensen-Shannon (JS) divergence (left), and softened reverse KL divergence $4\,\mathrm{KL}(\frac{1}{2}p + \frac{1}{2}q \,\|\, p)$ (right) as a function of model parameters. Lines show the progression of SGD-based JS training based on the original, saturating gradient and based on the alternative, non-saturating gradient (solid for learned critic; dotted for optimal critic). The original scheme converges to the JS divergence minimum. The alternative scheme, which by the results of this note is equivalent to a hybrid (SRKL, JS) scheme, converges to the SRKL divergence minimum as expected.

This is a valid approximation of the value, since $D_f(p, q) = \tilde{E}_f(p, q, d^*)$ for the optimal critic $d(x) = d^*(x) = \log p(x) - \log q(x)$. However the partial derivative of $\tilde{E}_f$ with respect to the parameters of q is not equal to the derivative of D_f with respect to the parameters of q. Thus it is not the case that optimizing $\tilde{E}_f$ using gradient descent can be straightforwardly related to optimizing D_f. This is the source of the discrepancy between our result and theirs.

11 Experimental validation of mathematical result

In order to validate our mathematical conclusions we conducted a simple experiment. Training behavior using the original and alternative gradients on a toy problem are shown in Figure 5. We see that the two cases minimize different divergences, as expected based on the theoretical arguments presented above.

REFERENCES

gradient descent gan optimization is locally stable.

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1): 131–142, 1966.

Martin Arjovsky and Lon Bottou. Towards principled methods for training generative adversarial networks. In *Proc. ICLR*, 2017.

Martin Arjovsky, Soumith Chintala, and Lon Bottou. Wasserstein generative adversarial networks. In *Proc. ICML*, pp. 214–223, 2017.

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. 2018.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. ICLR*, 2018.

Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 2007.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In *Proc. ICLR*, 2018.

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

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. ICLR*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? A large-scale study. In *Advances in neural information processing systems*, pp. 700–709, 2018.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of GANs. In *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. ICLR*, 2018.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

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

Wei Peng, Yuhong Dai, Hui Zhang, and Lizhi Cheng. Training GANs with centripetal acceleration. *arXiv preprint arXiv:1902.08949*, 2019.

Ben Poole, Alexander A Alemi, Jascha Sohl-Dickstein, and Anelia Angelova. Improved generator objectives for GANs. In *Proc. NIPS Workshop on Adversarial Training*, 2016.

Mark D Reid and Robert C Williamson. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12(Mar):731–817, 2011.

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised MAP inference for image super-resolution. In *Proc. ICLR*, 2017.

A Appendix

Extra details.