Adversarial Divergences are Good Task Losses for Generative Modeling

Gabriel Huang [1] **Gauthier Gidel** [1] **Hugo Berard** [1] **Ahmed Touati** [1] **Simon Lacoste-Julien** [1]

Abstract

Generative modeling of high dimensional data like images is a notoriously difficult and ill-defined problem. In particular, how to evaluate a learned generative model is unclear. In this paper, we argue that *adversarial learning*, pioneered with generative adversarial networks (GANs), provides an interesting framework to implicitly define more meaningful task losses for unsupervised tasks, such as for generating "visually realistic" images. By unifying GANs and structured prediction under the framework of statistical decision theory, we put into light links between recent advances in structured prediction theory and the choice of the divergence in GANs. We argue that the insights about the notions of "hard" and "easy" to learn losses can be analogously extended to adversarial divergences. We also discuss the attractive properties of adversarial divergences for generative modeling, and perform experiments to show the importance of choosing a divergence that reflects the final task.

1. Introduction

For structured prediction and data generation the notion of **final task** is at the same time crucial and not well defined. Consider machine translation; the goal is to predict a good translation, but even humans might disagree on the correct translation of a sentence. Moreover, even if we settle on a ground truth, it is hard to define what it means for a candidate translation to be close to the ground truth. In the same way, for data generation, the task of generating pretty pictures or more generally realistic samples is not well defined. Nevertheless, both for structured prediction and data generation, we can try to define **criteria** which characterize good solutions such as grammatical correctness for translation or non-blurry pictures for image generation. By incorporating enough criteria into a **task loss**, one can hope to approximate the final task, which is otherwise hard to formalize.

Supervised learning and structured prediction are well-defined problems once they are formulated as the minimization of such as task loss. The usual task loss in object classification is the generalization error associated with the classification error, or 0-1 loss. In machine translation, where the goal is to predict a sentence, a **structured loss**, such as the BLEU score (Papineni et al., 2002), formally specifies how close the predicted sentence is from the ground truth. The generalization error is defined through this structured loss. In both cases, models can be objectively compared and evaluated with respect to the task loss (i.e., generalization error). On the other hand, we will show that it is not as obvious in generative modeling to define a task loss that correlates well with the final task of generating realistic samples.

Traditionally in statistics, distribution learning is formulated as density estimation where the task loss is the expected negative-log-likelihood. Although log-likelihood works fine in low-dimension, it was shown to have many problems in high-dimension (Arjovsky et al., 2017). Among others, because the Kullback-Leibler is too strong of a divergence, it can easily saturate whenever the distributions are too far apart, which makes it hard to optimize.

In this work we give insights on how adversarial divergences (Liu et al., 2017) can be considered as task losses and how they address many problems of the KL by indirectly incorporating hard-to-define criteria. We define **neural adversarial divergences**[1] as the following :

$$\mathrm{Div}_{\mathrm{NN}}(p||q_\theta) \widehat{=} \sup_{\phi\in\Phi} \mathbf{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim p\otimes q_\theta}[\Delta(f_\phi(\boldsymbol{x}), f_\phi(\boldsymbol{x}'))] \quad (1)$$

where $\{f_\phi : \mathcal{X} \to \mathbb{R}^{d'} \ ; \ \phi \in \Phi\}$ is a class of parametrized neural networks, called the **discriminators** in the Generative Adversarial Network (GAN) framework (Goodfellow et al., 2014). The constraints Φ and the function $\Delta : \mathbb{R}^{d'} \times \mathbb{R}^{d'} \to \mathbb{R}$ determine properties of the resulting divergence. Using these notations, training a GAN can

[1]Université de Montréal. Correspondence to: Gabriel Huang <gbxhuang@gmail.com>.

Presented at the ICML 2017 Workshop on Principled Approaches to Deep Learning, Sydney, Australia, 2017.

[1]In this work we simply call them **adversarial divergences** since we are interested in GANs, but note that in a more general setting, any parametric discriminator could be considered.

be seen as training a generator network q_θ (parametrized by θ) to minimize the adversarial divergence $\mathrm{Div}_{\mathrm{NN}}(p||q_\theta)$, where the generator network defines the probability distribution q_θ over $\boldsymbol{x}$.

Our contributions are the following:

- We argue that compared to traditional divergences, adversarial divergences are a good compromise in terms of sample complexity, computation, ability to integrate prior knowledge, flexibility and ease of optimization.
- We unify structured prediction and generative adversarial networks using statistical decision theory, and show that they both amount to formalizing a final task into the minimization of a statistical task loss.
- We explain why it is necessary to choose a divergence that adequately reflects our final task in generative modeling. We make a parallel with results in structured learning (also dealing with high-dimensional data), which quantify the importance of choosing a good objective in a specific setting.
- We explore with some simple experiments how the properties of the discriminator transfer to the adversarial divergence. Our experiments suggest that adversarial divergences are especially adapted to problems such as image generation, where it is hard to formally define a perceptual loss that correlates well with human judgment.

2. Background

Here we briefly introduce the structured prediction framework because it can be related to generative modeling in some ways. We will later unify them formally, and present insights from recent theoretical results to choose a better divergence. We also unify adversarial divergences with traditional divergences in order to compare them in the next section.

2.1. Structured Prediction

The goal of structured prediction is to learn a classifier $h_\theta : \mathcal{X} \to \mathcal{Y}$ which predicts a structured output $\boldsymbol{y}$ from an input $\boldsymbol{x}$. The key difficulty is that $\mathcal{Y}$ usually has an exponential size[2] (e.g. it could be all possible sequence of symbols with a given length). Being able to handle this exponentially large set of outputs is one of the key challenges in structured prediction because it makes traditional multiclass classification methods unusable in general.[3] Standard practice in structured prediction (Taskar et al., 2003; Collins, 2002; Pires et al., 2013) is to consider predictors based on score functions $h_\theta(\boldsymbol{x}) \widehat{=} \arg\max_{\boldsymbol{y}' \in \mathcal{Y}} s_\theta(\boldsymbol{x}, \boldsymbol{y}')$, where $s_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, called the **score/energy function** (LeCun et al., 2006), assigns a score to each possible label $\boldsymbol{y}$ for an input $\boldsymbol{x}$. Typically, as in structured SVMs (Taskar et al., 2003), the score function is linear: $s_\theta(\boldsymbol{x}, \boldsymbol{y}) = \langle \theta, g(\boldsymbol{x}, \boldsymbol{y}) \rangle$, where $g(\cdot)$ is a predefined feature map. Alternatively, the score function could also be a learned neural network (Belanger & McCallum, 2016).

In order to evaluate the predictions objectively, we need to define a **task-dependent** structured loss $\ell(\boldsymbol{y}', \boldsymbol{y}; \boldsymbol{x})$ which expresses the cost of predicting $\boldsymbol{y}'$ for $\boldsymbol{x}$ when the ground truth is $\boldsymbol{y}$. We discuss the relation between the loss function and the actual final task in Section 4.2 . The goal is then to find a parameter θ which minimizes the generalization error:

$$\min_{\theta \in \Theta} \mathbf{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p} \left[\ell(h_\theta(\boldsymbol{x}), \boldsymbol{y}, \boldsymbol{x}) \right] \tag{2}$$

Directly minimizing (2) is often an intractable problem; this is the case when the structured loss ℓ is the 0-1 loss (Arora et al., 1993). Instead, the usual practice is to minimize surrogate losses $\mathcal{L}$ (Bartlett et al., 2006) which have nicer properties such as (sub-)differentiability or convexity, to get a tractable optimization problem:

$$\min_{\theta \in \Theta} \mathbf{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p} \left[\mathcal{L}(s_\theta(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}, \boldsymbol{y}) \right]. \tag{3}$$

If $\mathcal{L}$ is continuous and bounded from below then (3) is a well defined problem. The surrogate loss is said to be consistent (Osokin et al., 2017) when its minimizer is also a minimizer of the task loss.

A simple example of structured prediction task is machine translation. Suppose we want to translate French sentences to English; the input $\boldsymbol{x}$ is then a sequence of French words, and the output $\boldsymbol{y}$ is a sequence of English words belonging to a dictionary D with typically $|D| \approx 10000$ words. If we restrict the output sequence to be shorter than T words, then $|\mathcal{Y}| = |D|^T$, which is exponential. A desirable criterion is to have a translation with many words in common with the ground truth, which is typically enforced using BLEU scores to define the task loss.

2.2. Adversarial and Traditional Divergences

Because we will compare properties of adversarial and traditional divergences throughout this paper, we choose to first unify them with a formalism similar to Sriperumbudur et al. (2012); Liu et al. (2017):

$$\mathrm{Div}(p||q_\theta) \widehat{=} \sup_{f \in \mathcal{F}} \mathbf{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim p\otimes q_\theta} [\Delta(f(\boldsymbol{x}), f(\boldsymbol{x}'))] \tag{4}$$

Under this framework we give some examples of **traditional divergences**:

[2] Additionally, $\mathcal{Y}$ might depend on the input x, but we ignore this effect for clarity of exposition.

[3] Such as ones based on maximum likelihood.

- ψ-divergences with generator function ψ (which we call f-divergences) can be written in dual form (Nowozin et al., 2016)[4]

$$\text{Div}_\psi(p||q) \widehat{=} \sup_{f:\mathcal{X}\to\mathbb{R}} \mathbf{E}_{\boldsymbol{x}\sim p}[f(\boldsymbol{x})] - \mathbf{E}_{\boldsymbol{x}'\sim q_\theta}[\psi^*(f(\boldsymbol{x}'))] \tag{5}$$

where ψ^* is the convex conjugate. Depending on ψ, one can obtain any ψ-divergence such as the (reverse) Kullback-Leibler, the Jensen-Shannon, the Total Variation, the Chi-Squared[5].

- Wasserstein-1 distance induced by an arbitrary norm $\|\cdot\|$ (Sriperumbudur et al., 2012):

$$W(p||q) \widehat{=} \sup_{\substack{f:\mathcal{X}\to\mathbb{R} \\ \forall \boldsymbol{x}\in\mathcal{X}, \\ ||f'(\boldsymbol{x})||^*\leq 1}} \mathbf{E}_{\boldsymbol{x}\sim p}[f(\boldsymbol{x})] - \mathbf{E}_{\boldsymbol{x}'\sim q_\theta}[f(\boldsymbol{x}')] \tag{6}$$

which can be interpreted as the cost to transport all probability mass of p into q, where $\|\boldsymbol{x}-\boldsymbol{x}'\|$ is the unit cost of transporting $\boldsymbol{x}$ to $\boldsymbol{x}'$.

- Maximum Mean Discrepancy (Gretton et al., 2012):

$$\text{MMD}(p||q) \widehat{=} \sup_{\substack{f\in\mathcal{H} \\ ||f||_{\mathcal{H}}\leq 1}} \mathbf{E}_{\boldsymbol{x}\sim p}[f(\boldsymbol{x})] - \mathbf{E}_{\boldsymbol{x}'\sim q_\theta}[f(\boldsymbol{x}')] \tag{7}$$

where $(\mathcal{H}, K)$ is a Reproducing Kernel Hilbert Space induced by a Kernel $K(\boldsymbol{x},\boldsymbol{x}')$ on $\mathcal{X}$. The MMD has many interpretations in terms of moment-matching (Li et al., 2017).

In the optimization problems (5) and (6), whenever f is additionally constrained to be a neural network with specific architecture,[6] then we talk about **neural adversarial divergences**, or just **adversarial divergences** For instance, the adversarial Jensen-Shannon optimized in GANs corresponds to (5) with specific ψ (Nowozin et al., 2016), while the adversarial Wasserstein optimized in WGANs corresponds to (6) where f is a neural network. See (Liu et al., 2017) for interpretations and a review and interpretation of other divergences like the Wasserstein with entropic smoothing (Aude et al., 2016), energy-based distances (Li et al., 2017) which can be seen as adversarial MMD, and the WGAN-GP (Gulrajani et al., 2017) objective.

[4]The standard form is $\mathbf{E}_{\boldsymbol{x}\sim q_\theta}[\psi(\frac{p(x)}{q_\theta(x)})]$.

[5] For instance the Kullback-Leibler $\mathbf{E}_{x\sim p}[\log\frac{p(x)}{q_\theta(x)}]$ has the dual form $\sup_{f:\mathcal{X}\to\mathbb{R}} \mathbf{E}_{\boldsymbol{x}\sim p}[f(\boldsymbol{x})] - \mathbf{E}_{\boldsymbol{x}'\sim q_\theta}[\exp(f(\boldsymbol{x}')-1)]$. Some ψ require additional constraints, such as $||f||_\infty \leq 1$ for the Total Variation.

[6]In the nonparametric limit where all neural network architectures are considered, adversarial divergences degenerate to their traditional counterparts.

3. Why Should We Use the Adversarial Divergence?

We argue that adversarial divergences have many good properties which make them attractive for generative modeling. In this section, we compare adversarial divergences to traditional divergences in terms of statistical efficiency, computational cost, ability to integrate criteria related to the final task, and whether they constrain the form of the generator.

Divergence	Sample Comp.	Computation
f-Div (EXPL)	$O(1/\epsilon^2)$	MC, $O(n)$
f-Div (IMPL)	N/A	N/A
Wasserstein	$O(1/\epsilon^{d+1})$	Sinkhorn, $O(n^2)$
MMD	$O(1/\epsilon^2)$	analytic, $O(n^2)$
Adversarial	$O(p/\epsilon^2)$	SGD

Table 1. Statistical and Computational Properties of Divergences. EXPL and IMPL stand for explicit and implicit models (whether the density $q_\theta(x)$ can be computed).

3.1. Statistical and Computational Efficiency

Since we want to learn from finite data, we would like to know how well empirical estimates of a divergence approximate the population divergence, i.e., how many samples n we need to have with high probability that $|\text{Div}(p||q) - \text{Div}(\widehat{p}_n||\widehat{q}_n)| \leq \epsilon$, where $\epsilon > 0$, and $\widehat{p}_n, \widehat{q}_n$ are empirical distributions associated with p, q. Convergence rates for adversarial and traditional divergences are summarized in Table 1.

For explicit models which allow evaluating the density $q_\theta(x)$, one could use Monte-Carlo to evaluate the f-divergence with sample complexity $n = O(1/\epsilon^2)$, according to the Central-Limit theorem. However, with implicit models, one has to resort to the dual form (5), which typically requires approximating with a neural network, and thus becomes an adversarial f-divergence.

Adversarial divergences can be formulated as a classification/regression problem with a loss depending on the specific adversarial divergence. Therefore, they have a reasonable sample complexity of $O(p/\epsilon^2)$, where p is the VC-dimension/number of parameters of the discriminator (Arora et al., 2017), and can be solved using classic stochastic gradient methods.

A straightforward estimator of the Wasserstein is simply the Wasserstein distance between the empirical distributions $\widehat{p}_n$ and $\widehat{q}_n$, for which smoothed versions can be computed in $O(n^2)$ using specialized algorithms such as Sinkhorn's algorithm (Cuturi, 2013) or iterative Bregman projections (Benamou et al., 2015). However, the empirical Wasserstein estimator has sample complexity $n = O(1/\epsilon^{d+1})$ which is exponential in the number of dimen-

sions (see Sriperumbudur et al., 2012, Corollary 3.5). Thus the empirical Wasserstein is not a viable estimator in high-dimensions.

Maximum Mean Discrepancy admits an estimator with sample complexity $n = O(1/\epsilon^2)$, which can be computed analytically in $O(n^2)$. More details are given in the original MMD paper (Gretton et al., 2007). As the sample complexity is independent of the dimension of the data, one might believe that the MMD estimator behaves well in high dimensions. However, it was experimentally illustrated in Dziugaite et al. (2015) that MMD performs poorly for MNIST and Toronto face datasets, as the images generated have many artifacts and are clearly distinguishable from the training dataset. It was also shown theoretically in (Reddi et al., 2014) that the power of the MMD statistical test drops polynomially with increasing dimension.

Note that comparing divergences in terms of sample complexity can give good insights on what is a good divergence, but should be taken with a grain of salt as well. On the one hand, the sample complexities we give are upper-bounds, which means the estimators could potentially converge faster. On the other hand, one might not need a very good estimator of the divergence in order to learn in some cases. This is illustrated in our experiments with the empirical Wasserstein (Section 5) which has bad sample complexity but yields reasonable results.

3.2. Ability to Integrate Final Task

In Section 4, we will argue that in structured prediction, optimizing for the right task losses is more meaningful and can make learning considerably easier. Similarly in generative modeling, we would like divergences to integrate criteria that characterize the final task. We discuss that although not all divergences can easily integrate final task-related criteria, adversarial divergences provide a way to do so.

Pure f-divergences cannot directly integrate any notion of final task,[7] at least without tweaking the generator. The Wasserstein distance and MMD are respectively induced by a base metric $d(\boldsymbol{x}, \boldsymbol{x}')$ and a kernel $K(\boldsymbol{x}, \boldsymbol{x}')$. The metric and kernel give us the opportunity to specify a task by letting us express a (subjective) notion of similarity. However, the metric and kernel generally have to be defined by hand, as there is no obvious way to learn them end-to-end. For instance, Genevay et al. (2017) learn to generate MNIST by minimizing a smooth Wasserstein based on the L2-distance, while Dziugaite et al. (2015); Li et al. (2015) also learn to generate MNIST by minimizing the MMD induced by kernels obtained externally: either generic kernels based on the L2-distance or on autoencoder features. However, the results seems to be limited to simple datasets. Recently there has been a surge in doing MMD with kernel learning, with convincing results on LSUN, CelebA and Imagenet images. Mroueh et al. (2017) learn a feature map and try to match its mean and covariance, Li et al. (2017) learn kernels end-to-end, while Bellemare et al. (2017) do end-to-end learning of energy distances, which are closely related to MMD.

Adversarial divergences are defined with respect to a class of discriminators. Thus, changing the properties of the discriminator will likely affect the associated adversarial divergence. Additionally, the adversarial Wasserstein distance (Arjovsky et al., 2017) can also incorporate a custom metric. In Section 5 we give interpretations and experiments to assess the relation between the discriminator and the divergence.

3.3. Ease of Optimization and Stability

While adversarial divergences are learned and thus potentially much more powerful than traditional divergences, the fact that they are the solution to a hard, non-convex problem can make GANs unstable. Not all adversarial divergences are equally stable: Arjovsky et al. (2017) claimed that the adversarial Wasserstein gives more meaningful learning signal than the adversarial Jensen-Shannon, in the sense that it correlates well with the quality of the samples, and is less prone to mode dropping. We show experimentally on a simple setting that indeed the adversarial Wasserstein consistently give more meaningful learning signal than the adversarial Jensen-Shannon, regardless of the architecture discriminator (Section 5). Similarly to the WGAN, the MMD-GAN divergence (Li et al., 2017) was shown to correlate well with the quality of samples and to be robust to mode collapse.

3.4. Avoiding Generators with Special Structure

In some cases, imposing a certain structure on the generator yields a Kullback-Leibler divergence which involves some form of component-wise distance between samples, reminiscent of the Hamming loss (later defined in Section 4.3) used in structured prediction. However, doing maximum likelihood on generators having an imposed special structure can have drawbacks which we detail here. For instance, the generative model of a typical variational autoencoder can be seen as an infinite mixture of Gaussians (Kingma & Welling, 2013). The log-likelihood reduces to a reconstruction loss, i.e., a pixel-wise L2 distance between images analogous to the Hamming loss, which

[7] As pointed out by a reviewer, one could also attempt to induce properties of interest by adding a regularization term to the f-divergence. However, if we assume that maximum likelihood is itself often not a meaningful task loss, then there is no guarantee that minimizing a tradeoff between maximum likelihood and a regularization term is more meaningful or easier.

makes the training relatively easy and very stable. However, the Gaussians make the VAE unable to learn sharp distributions. Indeed it is a known problem that VAEs produce blurry samples (Arjovsky et al., 2017). Other examples are autoregressive models such as recurrent neural networks (Mikolov et al., 2010) which factorize naturally as $\log q_\theta(x) = \sum_i \log q_\theta(x_i|x_1, .., x_{i-1})$. Training using maximum likelihood results in teacher-forcing (Lamb et al., 2016): each ground-truth symbol is fed to the RNN, which then has to maximize the likelihood of the next symbol. Since teacher-forcing induces a lot of supervision, it is possible to learn using maximum-likelihood. Once again, there are similarities with the Hamming loss because each predicted symbol is compared with its associated ground truth symbol. However, among other problems, there is a discrepancy between training and generation. Sampling from q_θ would require iteratively sampling each symbol and feeding it back to the RNN, giving the potential to accumulate errors, which is not something that is accounted for during training. See Leblond et al. (2017) and references therein for more principled approaches to sequence prediction.

3.5. Sampling from Generator is Sufficient

Maximum-likelihood typically requires computing the density $q_\theta(x)$, which is not possible for implicit models such as GANs, from which it is only possible to sample. On the other hand, adversarial divergences can be estimated with reasonable sample complexity (see Section 3.1) only by sampling from the generator, without any assumption on the form of the generator. This is also true for MMD but generally not the case for the empirical Wasserstein, which has bad sample complexity as stated previously. Another issue of f-divergences such as the Kullback-Leibler or the Jensen-Shannon, is that they are either not defined or uninformative when p is not absolutely continuous w.r.t. q_θ (Nowozin et al., 2016), which makes them unusable for learning sharp distributions such as manifolds. On the other hand, some integral probability metrics, such as the Wasserstein, MMD, or their adversarial counterparts, are well defined for any distributions p and q_θ. In fact, even though the Jensen-Shannon is ill-defined for manifolds, the adversarial Jensen-Shannon used in the original GANs (Goodfellow et al., 2014) still allows learning realistic samples, even though the process is unstable (Salimans et al., 2016).

4. Choosing Better Task Losses

In this section, we try to provide insights in order to design the best adversarial divergence for our final task. After unifying structured prediction and generative adversarial networks, we review theoretical results on the choice of objectives in structured prediction, and discuss their interpretation in generative modeling.

4.1. Unifying Structured Prediction and Generative Adversarial Networks

We unify structured prediction and GANs using the framework of statistical decision theory. Assume that we are in a world with a set $\mathcal{P}$ of possible states and that we have a set $\mathcal{A}$ of actions. When the world is in the state $p \in \mathcal{P}$, the cost of playing action $a \in \mathcal{A}$ is the **(statistical) task loss** $L_p(a)$. The goal is to play the action minimizing the task loss.

Generative models with Maximum Likelihood. The set $\mathcal{P}$ of possible states is the set of available distributions $\{p\}$ for the data $\boldsymbol{x}$. The set of actions $\mathcal{A}$ is the set of possible distributions$\{q_\theta \ ; \ \theta \in \Theta\}$ for the model and the task loss is the negative log-likelihood,

$$L_p(\theta) \widehat{=} \mathbf{E}_{\boldsymbol{x}\sim p} \left[-\log(q_\theta(\boldsymbol{x})) \right] \tag{8}$$

Structured prediction. The set $\mathcal{P}$ of possible states is the set of available distribution $\{p\}$ for $(\boldsymbol{x}, \boldsymbol{y})$. The set of actions $\mathcal{A}$ is the set of prediction functions $\{h_\theta \ ; \ \theta \in \Theta\}$ and the task loss is the generalization error:

$$L_p(\theta) \widehat{=} \mathbf{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p} \left[\ell(h_\theta(\boldsymbol{x}), \boldsymbol{y}, \boldsymbol{x}) \right] \tag{9}$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ is a structured loss function.

GANs. The set $\mathcal{P}$ of possible states is the set of available distributions $\{p\}$ for the data $\boldsymbol{x}$. The set of actions $\mathcal{A}$ is the set of distributions $\{q_\theta \ ; \ \theta \in \Theta\}$ that the generator can learn, and the task loss is the adversarial divergence

$$L_p(\theta) \widehat{=} \sup_{f \in \mathcal{F}} \mathbf{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim p\otimes q_\theta} [\Delta(f(\boldsymbol{x}), f(\boldsymbol{x}'))] \tag{10}$$

Under this unified framework, the prediction function h_θ is analogous to the generative model q_θ, while the choice of the right structured loss ℓ can be related to the choice of the discriminators $\mathcal{F}$ which will induce a good adversarial divergence. We will further develop this analogy in Section 4.2.

4.2. Link Between Structured Losses and Adversarial Divergences

As discussed in the introduction, structured prediction and data generation involve a notion of **final task** which is at the same time crucial and not well defined. Nevertheless, for both we can try to define criteria which characterize good solutions. We would like the statistical task loss (introduced in Section 4.1), which corresponds to the generalization error in structured prediction, and the adversarial divergence in generative modeling, to incorporate task-related criteria. One way to do that is to choose a structured

loss that reflects the criteria of interest, or analogously to choose a class of discriminators, like a CNN architecture, such that the resulting adversarial divergence has good invariance properties. The whole process of building statistical task losses adapted to a final task, using the right structured losses or discriminators, is represented in Figure 1.

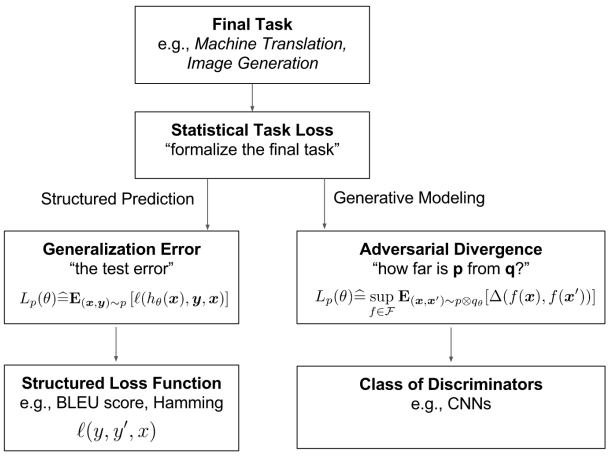


Figure 1. Formalizing a final task into the minimization of a statistical task loss.

For many prediction problems, the structured prediction community has engineered structured loss functions which induce **properties of interest** on the learned predictors. In machine translation, a commonly considered property of interest is for candidate translations to contain many words in common with the ground-truth; this has given rise to the BLEU score which counts the percentage of candidate words appearing in the ground truth. In the context of image segmentation, Osokin & Kohli (2014) have compared various structured loss functions which induces different properties on the predicted mask.

In the same vein as structured loss functions, adversarial divergences can be built to induce certain properties on the generated data. We are more concerned with generating realistic samples than having samples which are very similar with the training set; we actually want to extrapolate some properties of the true distribution from the training set. For instance, in the DCGAN (Radford et al., 2015), the discriminator has a convolutional architecture, which makes it potentially robust to small deformations that would not affect the visual quality of the samples significantly, while still making it able to detect blurry samples, which is aligned with our objective of generating realistic samples.

4.3. Some Task Losses are Easier to Learn

In this section we get insights from the convergence results of Osokin et al. (2017) in structured prediction. They show in a specific setting that some weaker structured loss functions are easier to learn than some stronger loss functions. In some sense, their results formalize the intuition in generative modeling that learning with weaker divergences is easier (Arjovsky et al., 2017) and more intuitive (Liu et al., 2017) than stronger divergences.

Intuition to Prove. Intuitively, strong losses such as the 0-1 loss are hard to learn because they do not give any flexibility on the prediction; the 0-1 loss only tells us whether a prediction is correct or not, and consequently does not give any clue about how close the prediction is to the ground truth. To get enough learning signal, we roughly need as many training examples as the number of possible outputs $|\mathcal{Y}|$, which is exponential in the dimension of y and thus inefficient. Conversely, weaker losses like the Hamming loss have more flexibility; because they tell us how close a prediction is to the ground truth, less example are needed to generalize well. The theoretical results proved by Osokin et al. (2017) formalize that intuition in a specific setting.

Theory to Back the Intuition. In a non-parametric setting (details in next paragraph), Osokin et al. (2017) formalize the intuition that weaker structured loss functions are easier to optimize. Specifically, they compare the 0-1 loss $\ell_{0-1}(\boldsymbol{y}, \boldsymbol{y}') \widehat{=} \mathbf{1}\left\{\boldsymbol{y} \neq \boldsymbol{y}'\right\}$ to the Hamming loss $\ell_{Ham}(\boldsymbol{y}, \boldsymbol{y}') \widehat{=} \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{\boldsymbol{y}_t \neq \boldsymbol{y}'_t\}$, when $\boldsymbol{y}$ decomposes as $T = \log_2 |\mathcal{Y}|$ binary variables $(y_t)_{1\leq t\leq T}$. They derive a worst case sample complexity needed to obtain a fixed error $\epsilon > 0$. For the 0-1 loss, they obtain a sample complexity of $O(|Y|/\epsilon^2)$ which is exponential in the dimension of y. However, for the Hamming loss, under certain constraints (see Osokin et al., 2017, section on exact calibration functions) they obtain a much better sample complexity of $O(\log_2 |Y|/\epsilon^2)$ which is polynomial in the number of dimensions, whenever certain constraints are imposed on the score function. Thus their results suggest that choosing the right structured loss might make training exponentially faster.

Limitations of the Theory. Although Osokin et al. (2017) give a lot of insights, their results must be taken with a grain of salt. Their analysis ignores the dependence on x and is non-parametric, which means that they consider the whole class of possible score functions for each given x. Additionally, they only consider convex consistent surrogate losses in their analysis, and they give upper bounds but not lower bounds on the sample complexity. It is possible that optimizing approximately-consistent surrogate losses instead of consistent ones, or making additional assumptions on the distribution of the data could yield better sample complexities.

Insights. Those theoretical results are consistent with our intuition that weaker losses are easier to optimize, and

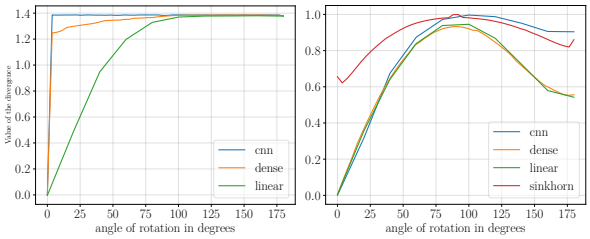
Figure 2. Images generated by the network after training with the Sinkorn-Autodiff algorithm on MNIST dataset (left) and CIFAR-10 dataset (right).

quantify in a specific setting how much harder it is to learn with strong structured loss functions like the 0-1 loss than weaker ones like the Hamming loss (here, exponentially harder).

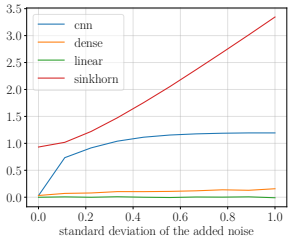
Relation with Adversarial Divergences. Under the unified framework we introduced in Section 4.2, choosing a structured loss function is analogous to choosing a class of discriminators to define a statistical task loss. In both cases (9) and (10) are designed to induce properties of interest on the solution of the optimization problem. Additionally, the fact that flexible statistical task losses, which can "smoothly" distinguish between good and bad models, are easier to optimize in the context of structured prediction, can be related to the belief that weaker adversarial divergences are easier to optimize in generative modeling. Arjovsky et al. (2017); Liu et al. (2017) compare traditional divergences in terms of strength, and give arguments why it might be easier to learn in weaker topologies than in stronger ones. For instance, distributions with disjoint support can be compared in weaker topologies like the Wasserstein but not in stronger ones like the Jensen-Shannon. They also show that the WGAN, based on the adversarial Wasserstein, is much more stable than the GAN, based on the adversarial Jensen-Shannon.

5. Experimental results

Importance of Sample Complexity. Since the sample complexity of the empirical Wasserstein (Section 3.1) is exponential in the dimension, we check experimentally whether training a generator to minimize the empirical Wasserstein distance fails in high dimensions. We implement the Sinkhorn-AutoDiff algorithm (Genevay et al., 2017) to compute the entropy-regularized L2-Wasserstein distance between minibatches of training images and generated images. Figure 2 shows generated samples after training with the Sinkhorn-Autodiff algorithm on both MNIST and CIFAR-10 dataset. On MNIST, the network manages to produce decent but blurry images. However, on CIFAR-10, which is a much more complex dataset, the network fails to produce meaningful samples, which would suggest that indeed the empirical Wasserstein should not be used for high-dimensional generative modeling.



(a) AdvJS (Rotations). (b) AdvW (Rotations).



(c) AdvW (Additive Noise).

Figure 3. **Top (a) and (b)**: divergences between MNIST and rotated MNIST. **Bottom (c)**: divergences between MNIST and noisy MNIST. AdvW plots for each model were rescaled, but using the same scaling factor across plots. When comparing different models/divergences, only the shape (but not the scale) of the curve matters, while for a same model the scale across different transformations does matter.

Robustness to Transformations. Intuitively, small rotations should not significantly affect the realism of images, while additive noise should. We study the effects of rotations and additive noise by plotting, for different amplitudes of transformation, various adversarial divergences between MNIST and rotated/noisy versions of it: three discriminators (linear, 1-layer-dense, 2-layer-cnn) combined with adversarial Jensen-Shannon (AdvJS) and adversarial Wasserstein (AdvW) formulations. Ideally, good divergences should vary smoothly (be robust) with respect to the amplitude of the transformation. For rotations (Figures 3a and 3b) and all discriminators except the linear, AdvJS saturates at its maximal value, even for small values of rotation, whereas the Wasserstein distance varies much more smoothly, which is consistent with Arjovsky et al. (2017). The fact that the linear AdvJS does not saturate for rotations shows that the architecture of the discriminator has a significant effect on the induced adversarial divergence, which confirms that there is a conceptual difference between the true JS and AdvJS, and even among different AdvJS. For additive Gaussian noise (Figure 3c), the linear discriminator is unable to distinguish the two distributions (it only sees the means of the distributions), whereas more complex architectures like CNNs do. In that sense the linear discriminator is too weak for the task, or not strict enough (Liu et al., 2017), which suggests that a better divergence involves trading off between robustness and strength.

Figure 4. Prototypes learned using linear discriminator (**left**), dense discriminator (**middle**), and CNN discriminator (**right**).

Learnability of Divergences. Since the final task is generative modeling, and because it is not easy to objectively compare divergences without talking of data generation, we design an experiment, inspired from the GAN framework, to evaluate the quality of samples learned using different divergences. We consider one of the simplest non-trivial generator one can think of,[8] a mixture of Diracs $q_\theta(\boldsymbol{x}) = \frac{1}{K}\sum_z \delta(\boldsymbol{x} - \boldsymbol{x}_z)$, parametrized by prototypes $\theta = (\boldsymbol{x}_z)_{1 \leq z \leq K}$. The generative process consists in sampling a discrete random variable $z \in \{1, ..., K\}$, and returning the prototype $\boldsymbol{x}_z$. We now train q_θ by minimizing various divergences, and compare the learned prototypes $(\boldsymbol{x}_z)$ in terms of quality and diversity of learned samples, which is our final task. Unlike typical GAN generators, our generator is so simple that any difference of quality in learned samples is likely to come from the divergence used to train it, and not from issues in optimizing the generator. We train mixtures of 100 prototypes by minimizing WGAN-GP (Gulrajani et al., 2017) divergences based on discriminators with linear, 1-hidden-layer dense, and CNN architectures (learned prototypes in Figure 4). The first observation is that the linear discriminator is too weak of a divergence: all prototypes only learn the mean of the training set. Now, the dense discriminator learns prototypes which sometimes look like digits, but are blurry or unrecognizable most the time. The samples from the CNN discriminator are never blurry and recognizable in the majority of cases. Our results suggest that indeed, even for simplistic models like a mixture of Diracs, using a CNN discriminator provides a better task loss for generative modeling of images.

6. Related Work

Most of the literature aiming at better understanding (adversarial) divergences and GANs has focused on specific issues, detailed below, which can make it hard to have a global view of adversarial divergences. In our paper we review those results and put them in perspective in an attempt to provide a more principled view of the nature and usefulness of adversarial divergences, with respect to traditional divergences. To the best of our knowledge, we are also the first to make a link between the generalization error of structured prediction and the adversarial divergence in generative modeling.

Work has been done to better understand the GAN objective in order to improve its stability Salimans et al. (2016). In particular Arjovsky et al. (2017) introduce the adversarial Wasserstein distance which makes training much more stable, and Gulrajani et al. (2017) improve the objective to make it more practical. There have been successive unifications of GAN objectives and divergences in order to gain better insights. For instance, Nowozin et al. (2016) generalize the GAN objective to any adversarial f-divergence. Sriperumbudur et al. (2012) unify traditional IPMs, analyze their statistical properties, and propose to view them as classification problems. However, the first papers to actually study the effect of restricting the discriminator to be neural network instead of any function are the MMD-GAN papers: Li et al. (2015); Dziugaite et al. (2015); Li et al. (2017); Mroueh et al. (2017); Bellemare et al. (2017) give interpretations of their framework in terms of moment matching. Liu et al. (2017) unify adversarial divergences with traditional divergences, introduce the notion of strong and weak divergence, and give a moment-matching interpretation of adversarial divergences. As for statistical properties, Arora et al. (2017) show that a optimal discriminator analysis does not really make sense, because the adversarial divergence relies on the limited capacity of the discriminator in order to generalize from finite data. Concerning theoretical understanding of learning in structured prediction, some recent papers are devoted to theoretical understanding of structured prediction such as (Cortes et al., 2016) and (London et al., 2016) which propose generalization error bounds in the same vein as Osokin et al. (2017) but with data dependencies.

7. Conclusion

We gave arguments in favor of using adversarial divergences rather than traditional divergences for generative modeling, the most important of which being the ability to account for the final task. After linking and unifying structured prediction and generative modeling under the framework of statistical decision theory, we interpreted recent results from structured prediction, and related them to the notions of strong and weak divergences. Moreover, viewing adversarial divergences as statistical task losses led us to believe that some adversarial divergences could be used as evaluation criteria in the future, replacing hand-crafted criteria which cannot usually be exhaustive. In some sense, we want to extrapolate a few desirable properties into a meaningful task loss. In the future we would like to investigate how to define meaningful evaluation criteria with minimal human intervention.

[8] A mixture of Diracs can be seen as a Gaussian Mixture Model with covariance matrices equal to zero.

Acknowledgments

This research was partially supported by the Canada Excellence Research Chair in "Data Science for Real-time Decision-making", by the NSERC Discovery Grant RGPIN-2017-06936 and by a Google Research Award.

References

Arjovsky, Martin, Chintala, Soumith, and Bottou, Léon. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Arora, Sanjeev, Babai, László, Stern, Jacques, and Sweedyk, Z. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pp. 724–733. IEEE, 1993.

Arora, Sanjeev, Ge, Rong, Liang, Yingyu, Ma, Tengyu, and Zhang, Yi. Generalization and equilibrium in generative adversarial nets (GANs). *arXiv preprint arXiv:1703.00573*, 2017.

Aude, Genevay, Cuturi, Marco, Peyré, Gabriel, and Bach, Francis. Stochastic optimization for large-scale optimal transport. *arXiv preprint arXiv:1605.08527*, 2016.

Bartlett, Peter L, Jordan, Michael I, and McAuliffe, Jon D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Belanger, David and McCallum, Andrew. Structured prediction energy networks. In *International Conference on Machine Learning*, pp. 983–992, 2016.

Bellemare, Marc G, Danihelka, Ivo, Dabney, Will, Mohamed, Shakir, Lakshminarayanan, Balaji, Hoyer, Stephan, and Munos, Rémi. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

Benamou, Jean-David, Carlier, Guillaume, Cuturi, Marco, Nenna, Luca, and Peyré, Gabriel. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Collins, Michael. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume (EMNLP)*, pp. 1–8, 2002.

Cortes, Corinna, Kuznetsov, Vitaly, Mohri, Mehryar, and Yang, Scott. Structured prediction theory based on factor graph complexity. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2514–2522, 2016.

Cuturi, Marco. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2292–2300, 2013.

Dziugaite, Gintare Karolina, Roy, Daniel M, and Ghahramani, Zoubin. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

Genevay, Aude, Peyré, Gabriel, and Cuturi, Marco. Sinkhorn-autodiff: Tractable wasserstein learning of generative models. *arXiv preprint arXiv:1706.00292*, 2017.

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

Gretton, Arthur, Borgwardt, Karsten M, Rasch, Malte, Schölkopf, Bernhard, Smola, Alexander J, et al. A kernel method for the two-sample-problem. *Advances in Neural Information Processing System (NIPS)*, 19:513, 2007.

Gretton, Arthur, Borgwardt, Karsten M, Rasch, Malte J, Schölkopf, Bernhard, and Smola, Alexander. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron. Improved training of wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lamb, Alex M, Goyal, Anirudh, Zhang, Ying, Zhang, Saizheng, Courville, Aaron C, and Bengio, Yoshua. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 4601–4609, 2016.

Leblond, Rémi, Alayrac, Jean-Baptiste, Osokin, Anton, and Lacoste-Julien, Simon. Searnn: Training rnns with global-local losses. *arXiv preprint arXiv:1706.04499*, 2017.

LeCun, Yann, Chopra, Sumit, Hadsell, Raia, Ranzato, M, and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 2006.

Li, Chun-Liang, Chang, Wei-Cheng, Cheng, Yu, Yang, Yiming, and Póczos, Barnabás. MMD GAN: Towards

deeper understanding of moment matching network. *arXiv preprint arXiv:1705.08584*, 2017.

Li, Yujia, Swersky, Kevin, and Zemel, Rich. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 1718–1727, 2015.

Liu, Shuang, Bousquet, Olivier, and Chaudhuri, Kamalika. Approximation and convergence properties of generative adversarial learning. *arXiv preprint arXiv:1705.08991*, 2017.

London, Ben, Huang, Bert, and Getoor, Lise. Stability and generalization in structured prediction. *Journal of Machine Learning Research*, 17(222):1–52, 2016.

Mikolov, Tomas, Karafiát, Martin, Burget, Lukas, Cernockỳ, Jan, and Khudanpur, Sanjeev. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 3, 2010.

Mroueh, Youssef, Sercu, Tom, and Goel, Vaibhava. Mcgan: Mean and covariance feature matching GAN. *arXiv preprint arXiv:1702.08398*, 2017.

Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-GAN: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709*, 2016.

Osokin, Anton and Kohli, Pushmeet. Perceptually inspired layout-aware losses for image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

Osokin, Anton, Bach, Francis, and Lacoste-Julien, Simon. On structured prediction theory with calibrated convex surrogate losses. *arXiv preprint arXiv:1703.02403*, 2017.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics (ACL)*, pp. 311–318, 2002.

Pires, Bernardo Avila, Szepesvari, Csaba, and Ghavamzadeh, Mohammad. Cost-sensitive multiclass classification risk bounds. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1391–1399, 2013.

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

Reddi, Sashank J, Ramdas, Aaditya, Póczos, Barnabás, Singh, Aarti, and Wasserman, Larry. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. *arXiv preprint arXiv:1406.2083*, 2014.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2234–2242, 2016.

Sriperumbudur, Bharath K, Fukumizu, Kenji, Gretton, Arthur, Schölkopf, Bernhard, Lanckriet, Gert RG, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Taskar, Ben, Guestrin, Carlos, and Koller, Daphne. Max-margin markov networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 25–32, 2003.