Adversarial training for predictive tasks: theoretical analysis and limitations in the deterministic case.

Anonymous Author(s)
Affiliation
Address
email

Abstract

To train a deep neural network to mimic the outcomes of processing sequences, a version of Conditional Generalized Adversarial Network (CGAN) can be used. It has been observed by others that CGAN can help to improve the results even for deterministic sequences, where only one output is associated with the processing of a given input. Surprisingly, our CGAN-based tests on deterministic geophysical processing sequences did not produce a real improvement compared to the use of an L_p loss; we here propose a first theoretical explanation why. Our analysis goes from the non-deterministic case to the deterministic one. It led us to develop an adversarial way to train a content loss that gave better results on our data.

1 Introduction

We consider the problem of mimicking a complicated processing sequence by learning some representation of the joint probability density function (pdf) that couples the outcomes of the sequence to its inputs. In geophysics, our field of application, processing sequences are usually based on workflows that represent a combination of algorithms and user-provided information to achieve a given task. Wave equation and signal processing are classical components of the algorithms, and geological priors are often part of user-provided information [1]. Several geophysical processing sequences aim at removing some undesired very structured events in the geophysical data [1], like the "ghost" events illustrated in Fig. 1. Learning an efficient representation that mimics such sequences can bring value, for example to take the best of various existing workflows, increase turnaround or obtain a processing guide. Deep Neural Networks (DNNs) provide a flexible tool to parameterize a function that predicts outcomes from inputs. Many explorations have recently been done using DNNs to mimic geophysical processing sequences, see for instance Refs. [2, 3, 4, 5, 6].

To train a DNN to predict outcomes from inputs, we may consider methods inspired from the Generative Adversarial Network (GAN) framework [7], in particular Conditional GAN (CGAN) [8, 9]. Indeed, CGAN can deal with joint pdfs (contrary to the original GAN formulation that deals only with single parameter pdfs), the originality being that the discriminator becomes conditioned by the input data [10, 4, 3, 6]. However, in the common context of a deterministic processing sequence, where only one outcome is generated when the processing sequence is applied to an input [2, 3, 4, 5, 6], using a simple L_p-norm based loss for the training usually gives good results [11]. So, can CGAN be pertinent even in the deterministic case? It has been observed that combining CGAN with a L_p loss may help to improve the results further, see e.g. Ref. [10] for natural image processing and Refs. [4, 6] for geophysical processing.

Surprisingly, our Wasserstein CGAN-based trainings [12] on deterministic geophysical processing sequences, like the "deghosting" (or ghost removal) sequence [13], did not help to produce a real improvement in our tests compared to the use of a L_p loss, see Fig. 2. In this paper, we propose a theoretical analysis of this aspect. First, we remind why L_p losses should perform well

Submitted to 1st I Can't Believe It's Not Better Workshop (ICBINB@NeurIPS 2020). Do not distribute.

in the deterministic prediction case. Then, we point out from the Wasserstein point of view what CGAN should bring compared to a L_p loss, taking the opportunity to discuss the Wasserstein CGAN foundations. Our analysis gives an explanation of why CGAN may perform more poorly than expected, and also leads to a proposal of an adversarial way to train a content loss that we call "Content CGAN" (C-CGAN); it gave better results on our data as illustrated in Fig. 2. For completeness, we start all our theoretical considerations from the non-deterministic prediction case, where multiple outcomes related to one input are possible, and then take the deterministic limit.

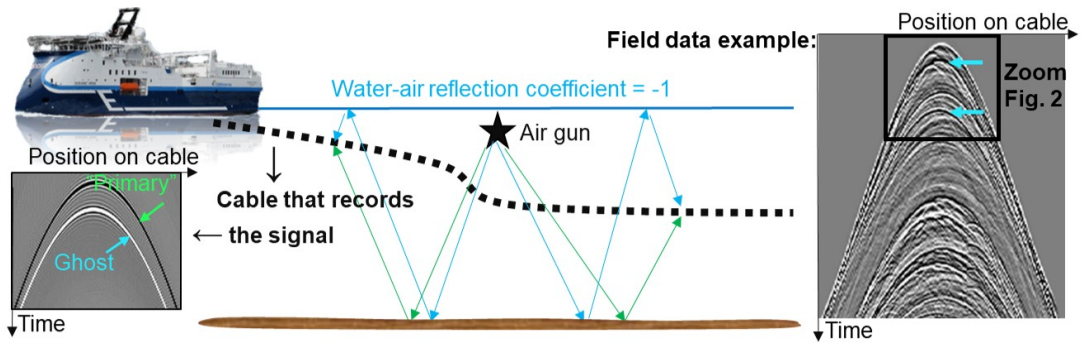

Figure 1: Marine seismic data acquisition. The pressure wavefield generated by an airgun is reflected in the subsurface, then comes back to the surface and is recorded along a cable pulled by a boat. Billions of data over thousands of square kilometers are recorded. A particularity of the geophysical data is to consist of very structured and continuous events, corresponding to discontinuities (layers) in the subsurface. The greyscale represents the polarity of the wavefield (black: positive, white: negative). Blue highlighted events have reflected on the water surface and are called "ghosts"; they look like "duplicate" events with reverse polarity; they interfere with the other events and must be removed by a "deghosting" sequence for some further applications.

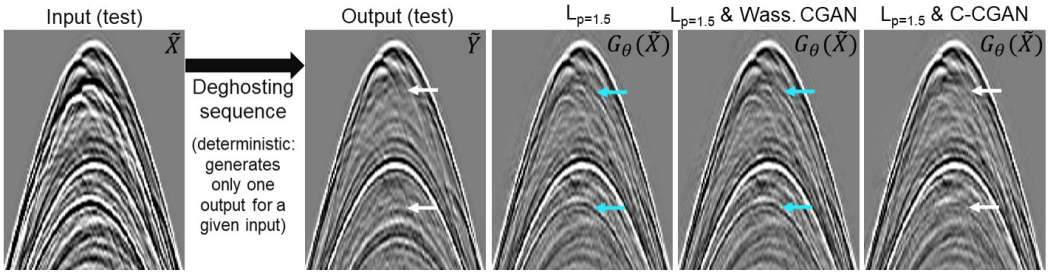

Figure 2: The DNN training data consists in 200 randomly extracted input "images" of size $564 \times 551 \times 1$, representing 0.001% of the total field data, together with corresponding output images generated by a conventional deghosting sequence. On the left, a conventional deghosting result is shown on a test data (chosen "far" from the training data). On the right, various DNN predictions from the input test data, shown before training full convergence to highlight the differences (40 epochs). Ghost residuals (blue arrows) can be observed on L_p and adding Wasserstein CGAN does not improve the result. Adding our C-CGAN more satisfyingly removes the ghost residuals (white arrows). However, the main benefit of C-CGAN is to accelerate the training as at full convergence (150 epochs) the difference between C-CGAN and L_p becomes quite smaller.

2 Notations

$\mathcal{X}$ denotes the input data space and $\mathcal{Y}$ the output data space. $P_{Y,X} = P_X P_{Y|X}$ denotes the joint pdf associated to the (possibly non-deterministic) processing sequence we wish to mimic. P_X is the marginal pdf that describes the distribution of the input data; realizations of the random variable $X \sim P_X$ are denoted by $\tilde{X} \in \mathcal{X}$. $P_{Y|X}$ is the conditional pdf that describes the outcomes of the processing sequence related to a given input; realizations of the random variable $Y \sim P_{Y|X}$ are

denoted by $\tilde{Y} \in \mathcal{Y}$. $G_\theta^Z : \mathcal{X} \to \mathcal{Y}$ represents a prediction function parameterized by a model θ, here a DNN, where the latent space random variable $Z \sim P_Z$ gives the flexibility to produce multiple outcomes related to a given input (for the non-deterministic prediction case). θ is to be optimized so that the Z-realizations of $G_\theta^Z(X)$ tend to mimic the realizations of $Y \sim P_{Y|X}$. $\mathbb{E}$ denotes the expectation over a specified random variable.

The deterministic prediction limit can be taken considering both:

- The "empirical" joint pdf, for instance, for $P_{Y,X}(\tilde{Y}, \tilde{X}) \to \frac{1}{N_D} \sum_{i=1}^{N_D} \delta(\tilde{Y} - \tilde{Y}_i)\delta(\tilde{X} - \tilde{X}_i)$. $\{\tilde{X}_i, \tilde{Y}_i; i = 1..N_D\}$ denotes a set of input and output data realizations.
- G_θ^Z independent of Z, so that a unique outcome is predicted by the DNN for each input.

3 Which processing sequences are suitable for the use of a L_p loss?

L_p losses are defined for $p \geq 1$ by

$$C_{L_p}^p(G_\theta^Z) \quad = \quad \mathbb{E}_{Z \sim P_Z} \mathbb{E}_{(Y,X) \sim P_{Y,X}} ||Y - G_\theta^Z(X)||_{L_p}^p, \tag{1}$$

where the image space L_p-norm is defined by

$$||\tilde{Y} - \tilde{Y}^{(2)}||_{L_p}^p = \int_\Omega \left| \tilde{Y}(y) - \tilde{Y}^{(2)}(y) \right|^p d\mu(y), \quad \forall (\tilde{Y}, \tilde{Y}^{(2)}) \in \mathcal{Y} \times \mathcal{Y}. \tag{2}$$

Each realization $\tilde{Y}$ represents an image indexed by the positions y in the output "pixel space" Ω, i.e. $\tilde{Y}(y)$, and $\mu(y)$ represents the counting measure (we introduce it for notational simplicity and generality; if $y \in \Omega$ is considered continuous, $\mu(y)$ should be replaced by the Lebesgue measure).

Training aims to minimize $C_{L_p}^p(G_\theta^Z)$, eq. (1), with respect to θ. As $C_{L_p}^p$ measures a similarity between one realization of Y and one realization of $G_\theta^Z(X)$, the trained prediction function G_θ^Z will tend to become independent of Z and output some "average" of all outcomes related to one input data. Indeed, we can easily compute the optimum for $p = 2$: $G_\theta^Z(X) \approx \mathbb{E}_{Y \sim P_{Y|X}} Y$, and for $p = 1$: $G_\theta^Z(X) \approx \mathbb{M}_{Y \sim P_{Y|X}} Y$ where $\mathbb{M}$ denotes the median [11].

In the non-deterministic prediction case, if the multiple outcomes are related to structured events, training with an L_p loss is obviously not suitable as it would tend to produce blurry predictions (due to the "averaging"). However, if the multiple outcomes are related to zero-"average" noise, an L_p loss is suitable and would even tend to produce denoised predictions. Consequently, a L_p loss is also suitable to the deterministic prediction case. A denoising effect can still occur if each of the single outcomes are affected by zero-"average" noise. Note that $p = 1$ (the median) is more robust to outliers than $p = 2$ but harder to train. $p = 1.5$ has been chosen in Fig. 1, representing a current compromise in geophysics [1].

This being posed, what could CGAN bring compared to a L_p loss in the deterministic case? Let us first discuss the CGAN foundations in the general non-deterministic case, from the Wasserstein point of view and complementarily to Ref. [12], and then analyze the deterministic limit.

4 Wasserstein CGAN for processing sequences

4.1 Non-deterministic prediction case

For notational purposes, let us consider the "parameterized" conditional pdf $P_{Y|X}^{(par)}$ whose realizations correspond to the ones of $G_\theta^Z(X)$. In other words, for any function D, $P_{Y|X}^{(par)}$ is defined so that

$$\mathbb{E}_{Y^{(par)} \sim P_{Y|X}^{(par)}} D(Y^{(par)}) = \mathbb{E}_{Z \sim P_Z} D(G_\theta^Z(X)), \tag{3}$$

where we keep the superscript $^{(par)}$ to make explicit which random variable is related to the parameterized pdf. Note that imposing a gaussian parameterization to $P_{Y|X}^{(par)}$ and using cross-entropy (XE) as a loss leads to eq. (1), as recalled in Appendix A. This allows us to understand from another point

of view the conclusions of §3: L_p losses are suited when the outcomes follow gaussian statistics, and are not suited when the gaussian assumption is too simplistic (as often with structured outcomes).

We now wish to define a similarity measure between the two joint pdfs $P_{Y,X}$ and $P_{Y,X}^{(par)} = P_X P_{Y|X}^{(par)}$, without having to consider any explicit parameterization like the gaussian one. XE is not adapted and Wasserstein distances seem like a natural choice. However, the original Wasserstein formulation is suited to single parameter pdfs or even conditional pdfs [14] but not to joint pdfs. We propose the following straightforward generalization ($p \geq 1$ and $r \geq 1$):

$$\begin{aligned} JW_{L_r}(P_{Y,X}, P_{Y,X}^{(par)}) &= \mathbb{E}_{X\sim P_X} W_{L_r}(P_{Y|X}, P_{Y|X}^{(par)}) \\ W_{L_r}(P_{Y|X}, P_{Y|X}^{(par)}) &= \Big(\inf_{\Pi^X_{Y,Y^{(par)}}} \mathbb{E}_{(Y,Y^{(par)})\sim \Pi^X_{Y,Y^{(par)}}} ||Y - Y^{(par)}||^p_{L_r} \Big)^{\frac{1}{p}}. \end{aligned} \tag{4}$$

$P_{Y|X}$ and $P_{Y|X}^{(par)}$ are considered as single parameter pdfs for each realization of X, and W_{L_r} represents a p-Wasserstein distance between $P_{Y|X}$ and $P_{Y|X}^{(par)}$ for any L_r-norm choice [14]. The infimum is taken over all joint pdfs $\Pi^X_{Y,Y^{(par)}}$ with marginals $P_{Y|X}$ and $P_{Y|X}^{(par)}$, X being considered as a parameter. Then, the expectation over all realizations of X is taken to obtain JW_{L_r}, representing a distance between the joint pdfs $P_{Y,X}$ and $P_{Y,X}^{(par)}$, as demonstrated in Appendix B. Switching to the dual formulation and taking $p = 1$ allows to simplify the second line of eq. (4) into the Kantorovitch-Rubinstein (KR) formulation [14, 15]

$$W_{L_r}(P_{Y|X}, P_{Y|X}^{(par)}) = \sup_{||D_X||_{Lip_{L_r}} \leq 1} \Big[\mathbb{E}_{Y\sim P_{Y|X}} D_X(Y) - \mathbb{E}_{Y^{(par)}\sim P_{Y|X}^{(par)}} D_X(Y^{(par)}) \Big]. \tag{5}$$

Note that the "discrimator" $D_X : \mathcal{Y} \to \mathbb{R}$ is parameterized by the input data. It is constrained to be 1-Lipschitz for the L_r-norm, i.e. $||D_X||_{Lip_{L_r}} \leq 1$, the "Lipschitz-norm" being defined by $||D_X||_{Lip_{L_r}} = \sup_{\tilde{Y}\neq\tilde{Y}^{(2)}} \frac{|D_X(\tilde{Y}) - D_X(\tilde{Y}^{(2)})|}{||\tilde{Y}-\tilde{Y}^{(2)}||_{L_r}}, \forall(\tilde{Y}, \tilde{Y}^{(2)}) \in \mathcal{Y} \times \mathcal{Y}$ [14, 15]. As demonstrated in Appendix C, if $D_X(\tilde{Y})$ is differentiable with respect to $\tilde{Y}$ with a continuous derivative, the Lipschitz-norm simplifies into

$$||D_X||_{Lip_{L_r}} = \sup_{\tilde{Y}\in\mathcal{Y}} \Big(\int_\Omega \Big| \frac{\partial D_X(\tilde{Y})}{\partial \tilde{Y}(y)} \Big|^s d\mu(y) \Big)^{\frac{1}{s}} \quad \text{with} \quad 1/r + 1/s = 1. \tag{6}$$

Inserting eq. (3) into eq. (5) to come back to G_θ^Z, we finally obtain the following tractable form

$$JW_{L_r}(G_\theta^Z) = \mathbb{E}_{X\sim P_X} \sup_{||D_X||_{Lip_{L_r}} \leq 1} \Big[\mathbb{E}_{Y\sim P_{Y|X}} D_X(Y) - \mathbb{E}_{Z\sim P_Z} D_X(G_\theta^Z(X)) \Big]. \tag{7}$$

Eqs. (6) and (7) provide an adversarial training framework for predictive tasks: JW_{L_r} contains a supremum principle on D_X, but the result is to be minimized with respect to θ during the training.

The discriminator necessary dependency on the input data $\tilde{X} \in \mathcal{X}$ establishes the relation with CGAN [8, 9] and its Wasserstein counterpart [12], that is well known. However, we underline some formal points that were not discussed in previous works to our knowledge:

- This dependency can possibly be very strong and discontinuous, leading in the most general case to one different discriminator per input data in eq. (7). Of course, this would be inefficient numerically and is usually not necessary (especially when images lie in low dimensional manifolds so that they do not vary rapidly). However, the considerations in this section lead to some clarification on the possibility of using many different discriminators within Wasserstein CGAN if needed.
- Eq. (6) provides the generalization to $L_{r\neq 2}$-norms to the derivative-based Lipschitz constraint of Ref. [16].
- We established that JW_{L_r} represents a distance between two joint pdfs with same marginal.

We mentioned in §3 that training with a L_p loss, eq. (1), compares one realization of $P_{Y|X}$ to one Z-realization of $G_\theta^Z(X)$, which leads to "averaging". Training with JW_{L_r}, eq. (7), compares all

realizations of $P_{Y|X}$ to all Z-realizations of $G_\theta^Z(X)$, i.e. G_θ^Z can learn to mimic the realizations of $P_{Y|X}$ and no "averaging" occurs. This fundamental difference would help to produce unblurred results in the non-deterministic case, when multiple outcomes are related to structured events. In the deterministic prediction case, however, what JW_{L_r} would bring compared to $C_{L_p}^p$ is unclear. This is what we discuss now.

4.2 Advantages of Wasserstein CGAN in the deterministic case?

To take the deterministic prediction case limit, we use the method mentioned in §2. Eq. (7) becomes

$$JW_{L_r}(G_\theta) = \sum_{i=1}^{N_D} \sup_{||D_{\tilde{X}_i}||_{Lip_{L_r}} \leq 1} \Big[D_{\tilde{X}_i}(\tilde{Y}_i) - D_{\tilde{X}_i}(G_\theta(\tilde{X}_i)) \Big], \tag{8}$$

where $\tilde{X}_i$ and $\tilde{Y}_i$ denote input and output data pairs. We use a discriminator architecture form

$$D_{\tilde{X}_i}(\tilde{Y}) = \int_\Omega F(\tilde{X}_i, \tilde{Y})(y) d\mu(y), \tag{9}$$

where $F(\tilde{X}_i, \tilde{Y}) \in \mathcal{Y}$ is parameterized by a convolutional DNN without striding and an additional last "layer" simply represents a sum over the pixels. The last layer is equivalent to a global average pooling [9, 17] and has the advantage to make the discriminator DNN model independent of the size of the data (i.e. the model can be used for any data size). This architecture allows for an interpretation of what the discriminator learns since $F(\tilde{X}_i, \tilde{Y})$ lies in the output image space. Also, in our tests on geophysical processing tasks, it led to the highest Wasserstein distance estimates (or supremum values) when training the discriminator. So, it is the architecture we choose.

Just to gain insight, we first consider a linear parameterization $F(\tilde{X}_i, \tilde{Y})(y) = \alpha(\tilde{X}_i)(y)\tilde{Y}(y)$, where α is a function of $\tilde{X}_i$ parameterized by a DNN. Inserting this in eqs. (6) and (8), we obtain [1]

$$JW_{L_r}(G_\theta) \rightarrow \sum_{i=1}^{N_D} \sup_{||\alpha(\tilde{X}_i)||_{(s)}=1} \int_\Omega |\alpha(\tilde{X}_i)(y)| \times \Big| \tilde{Y}_i(y) - G_\theta(\tilde{X}_i)(y) \Big| d\mu(y) \tag{10}$$

$$||\alpha(\tilde{X}_i)||_{(s)} = \Big(\int_\Omega |\alpha(\tilde{X}_i)(y)|^s d\mu(y) \Big)^{\frac{1}{s}} \quad \text{with} \quad 1/r + 1/s = 1.$$

Compared to the L_1-based loss defined by eqs. (1) and (2) with $p = 1$, i.e.

$$C_{L_1}(G_\theta) \quad = \quad \sum_{i=1}^{N_D} \int_\Omega \Big| \tilde{Y}_i(y) - G_\theta(\tilde{X}_i)(y) \Big| d\mu(y), \tag{11}$$

eq. (10) adds learnt positive weights $|\alpha(\tilde{X}_i)(y)|$ with norm equal to one. In other words, with a simple linear parameterization for F in the deterministic case, Wasserstein CGAN only adds automatic learning of optimal data-dependent variance-like weights compared to a L_1 loss. In configurations where adding variance-like weights would not affect the position of most of the minimums in the loss valley, the use of eq. (10) instead of a L_p loss would not produce a strong prediction improvement once the training convergence has been reached; the main effect should be to improve the convergence speed and possibly deal slightly better with the amplitude and the noise present in the data.

What about more involved parameterizations for $F(\tilde{X}_i, \tilde{Y}_i)$, using a convolutional DNN and non-linear activations? This will produce more involved transformations of $\tilde{Y}_i$ and $G_\theta(\tilde{X}_i)$ than a simple reweighting. Indeed, $\tilde{Y}_i \rightarrow F(\tilde{X}_i, \tilde{Y}_i)$ and $G_\theta(\tilde{X}_i) \rightarrow F(\tilde{X}_i, G_\theta(\tilde{X}_i))$ would then correspond to a preconditioning of the outputs. The supremum principle allows to learn the preconditioning that concentrates on the less matched events, i.e. that makes JW_{L_r} the most sensitive to the

[1] We have $JW_{L_r} \rightarrow \sum_{i=1}^{N_D} \sup_{||\alpha(\tilde{X}_i)||_{(s)} \leq 1} \int_\Omega \alpha(\tilde{X}_i)(y) \Big(\tilde{Y}_i(y) - G_\theta(\tilde{X}_i)(y) \Big) d\mu(y)$, that necessarily leads to the supremum argument $\alpha^{sup}(\tilde{X}_i)(y) = |\alpha^{sup}(\tilde{X}_i)(y)| \times \text{sign}\Big(\tilde{Y}_i(y) - G_\theta(\tilde{X}_i)(y) \Big)$ and a saturation of the constraint. Note that the dependency of α on $\tilde{X}_i$ is sufficient to define the sign as only one $\tilde{Y}_i$ is associated to $\tilde{X}_i$ in the deterministic case.

differences between the prediction and the output data. Still, in configurations where adding such a preconditioning would not affect the position of most of the minimums in the loss valley, the main effect should be to improve the training convergence and possibly to deal better with the amplitude and noise present in the output data. These situations may tend to occur amongst others when gross data amplitudes vary not too rapidly (thus also the variance-like weights or preconditionings), like in the case of Fig. 2. This is an element to understand why our Wasserstein CGAN trainings on deterministic geophysical processing sequences did not help to produce a real improvement.

Another element is that the method contains free parameters. Two of these are related to the Lipschitz constraint: s in eq. (6), and a weight to impose the constraint for instance using the method of Ref. [16]. Also, like in Refs. [10, 4, 6], we observed JW_{L_r} has to be combined to a L_p loss to give correct results, thus an additional weight is needed. We tuned the latter so that JW_{L_r} and L_p tend to contribute equally and used $s = 1$ (that represented a good compromise) to obtain the results in Fig. 2. However, as these hyper-parameters are data dependent, it may explain why Wasserstein CGAN did not produce a systematic improvement in our tests. The question of tuning these parameters the best for any kind of data is important but goes beyond the scope of this paper and is left for a future study.

4.3 Content CGAN: An adversarial way to train a content loss

Another difficulty with Wasserstein CGAN is that it is not feasible to resolve exactly the supremum principle in JW_{L_r}, eq. (8), at each iteration. This can lead to possible instabilities in the G_θ training and contribute to the explanation of some poor results. In this section, we propose to tackle a part of this specific problem by a heuristic reformulation of eq. (8). Note that the linearized case result of §4.2 can equivalently be recovered by firstly imposing the following form to F

$$F(\tilde{X}_i, \tilde{Y})(y) \rightarrow F(\tilde{X}_i, \tilde{Y})(y) \times \text{sign}\Big(F(\tilde{X}_i, \tilde{Y}_i)(y) - F(\tilde{X}_i, G_\theta(\tilde{X}_i))(y)\Big), \tag{12}$$

and then do the linearized approximation (remember footnote 1 and beware that the argument of the sign does not depend on $\tilde{Y}$ but on $\tilde{Y}_i$, which is important for the Lipschitz-norm, eq. (6)). Keeping this form for any non-linear parameterization of $F(\tilde{X}_i, \tilde{Y})$ and inserting eq. (12) in eq. (8), we obtain

$$JW_{L_r}(G_\theta) \quad \rightarrow \quad \sum_{i=1}^{N_D} \sup_{||D_{\tilde{X}_i}||_{Lip_{L_r}} \leq 1} \int_\Omega \Big| F(\tilde{X}_i, \tilde{Y}_i)(y) - F(\tilde{X}_i, G_\theta(\tilde{X}_i))(y) \Big| d\mu(y), \tag{13}$$

where $D_{\tilde{X}_i}$ is defined through eqs. (9) and (12). Eq. (13) looks like a L_1-based content loss [10] that would adversarially be trained, simultaneously with the G_θ training. A good content loss should tend to maximize the differences between a prediction that has been "preconditioned" (through the DNN F) and the corresponding similarly preconditioned output data. This is achieved by the supremum principle in eq. (13), where the Lipschitz constraint provides robustness (to avoid singularities...). This heuristical reasoning leads to our "Content CGAN" (C-CGAN) loss. Among other advantages, the C-CGAN loss always remains positive, even if the supremum principle is not well resolved at some iteration, whereas the Wasserstein CGAN loss, eq. (8), might not.

4.4 DNN architectures and results

In the results presented in Fig. 2, the DNN inputs a ghosted image $\tilde{X}$ and predicts a deghosted image $G_\theta(\tilde{X})$. The prediction function G_θ architecture is Unet inspired [18]. The $F(\tilde{X}, \tilde{Y})$ architecture, that defines the discriminator through eq. (9), is Denet inspired [19]. For the $\tilde{X}$-dependency of F, we found it sufficient to concatenate $\tilde{X}$ to the first Denet layer; however, this certainly would deserve a specific study for the reason underlined in §3. As mentioned in §4.2, we used $s = 1$ in the Lipschitz-norm, eq. (6), and combined Wasserstein CGAN (or C-CGAN) to a L_p loss, so that both contributions contribute equally. Fig. 2 shows a result at 40 epochs, before full convergence (150 epochs). We see that C-CGAN gave better results than CGAN or L_p only on our data. However, the main benefit of C-CGAN is to accelerate the training. At full convergence we observed the differences between C-CGAN and L_p become quite smaller, certainly for the reason outlined in §4.2.

Fig. 3 proposes a way to demystify what the discriminator has learnt and interpret the interest of C-CGAN. Firstly through $F(\tilde{X}_i, \tilde{Y}_i)(y)$, defined in the output image space $\mathcal{Y}$. It shows that C-CGAN learns to concentrate on the most important events, i.e. around the ghosts; this is satisfying and contributes to explain why C-GCAN achieves better deghosting more rapidly.

Secondly, we consider so-called "adjoint-input" $\frac{\partial LOSS(G)}{\partial G(y)}\big|_{G=G_\theta(\tilde{X}_i)}$, which is back-propagated in G_θ to compute how to update θ [20]. The adjoint-input is also defined in $\mathcal{Y}$ and allows for a visualization of the areas where the events should be better predicted after the update. Fig. 3 shows that the L_p adjoint-input tends to "put the weight" on all areas, regardless of their relative importance, thus not to concentrate more specifically on the ghost areas. The Wasserstein CGAN adjoint-input also tends to concentrate on many areas, contributing to explain why it brought no improvement in Fig. 2. The texture of this adjoint-input may seem atypical; we verified that the optimization of eq. (8) and of G_θ converged, but further analysis regarding the hyper-parameters mentioned in §4.2 is on the way. The C-CGAN adjoint-input, however, learned to concentrates more specifically around the ghost areas, contributing to explain why it converges more rapidly towards an acceptable solution in Fig. 2.

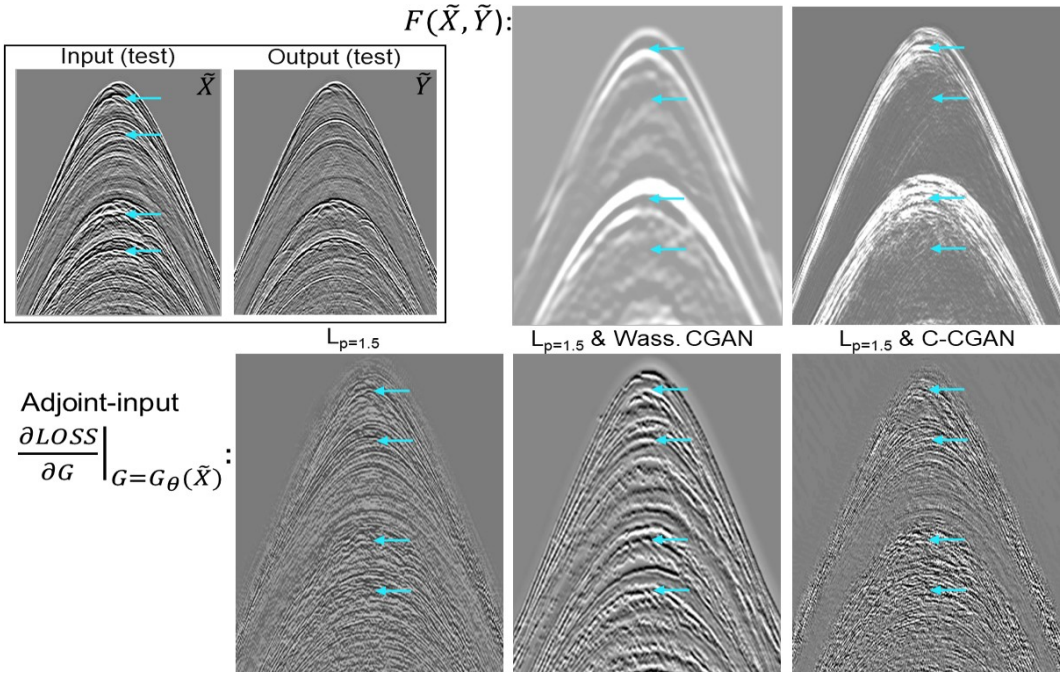

Figure 3: Still for the deghosting task and the same data than in Fig. 2, this figure illustrates what the discriminator learns from the $F(\tilde{X},\tilde{Y})$ (top) and adjoint-input (bottom) point of views. We observe that Wasserstein CGAN and L_p only tend to "put a weight" on many areas, regardless of their relative importance, while C-CGAN tends to learn to concentrate more on the important areas for deghosting, i.e. around the blue arrows.

5 Conclusion and future work

In this paper, we proposed a theoretical analysis of CGAN for predictive tasks. We took the opportunity to discuss the CGAN foundations from the Wasserstein point of view, and pointed out what CGAN should bring compared to a L_p loss in the deterministic prediction case. We explained that Wasserstein CGAN may perform more poorly than expected when the corresponding data-space "preconditioning" does not affect the position of most of the minimums in the loss valley (for instance when gross data amplitudes vary not too rapidly), or due to a difficulty with automatically tuning the corresponding hyper-parameters. Another difficulty is that the Wasserstein CGAN loss represents a distance only if the supremum principle is perfectly resolved numerically; our C-CGAN formalism helps to overcome this and gives better results on our data.

This first analysis is still to be confirmed by further studies. It is certainly data dependent (geophysical data being specific, with very structured and continuous events). Among others, understanding "physically" how to tune the CGAN hyperparameters for any kind of data is important in the deterministic as well as in the non-deterministic prediction cases; this will be a future study.

Broader Impact

Learning an efficient representation that mimics involved processing sequences can bring value in a general industrial context, not only in geophysics. The goal can be to take the best of various existing workflows, increase turnaround or obtain a processing guide.

References

[1] O. Yilmaz, *Seismic data analysis*. Society of Exploration Geophysicists, 2001.

[2] S. Alwon, "Generative adversarial networks in seismic data processing," *SEG 88th annual meeting*, 2018.

[3] F. Picetti, V. Lipari, P. Bestagini, and S. Tubaro, "A generative adversarial network for seismic imaging applications," *SEG 88th annual meeting*, 2018.

[4] A. Halpert, "Deep learning-enabled seismic image enhancement," *SEG 88th annual meeting*, 2018.

[5] X. Si, Y. Yuan, F. Ping, Y. Zheng, and L. Feng, "Ground roll attenuation based on conditional and cycle generative adversarial networks," *SEG Workshop: Traditional vs. Learning*, 2019.

[6] Z. Zhang and Y. Lin, "Data-driven seismic waveform inversion: A study on the robustness and generalization," *IEEE Transactions on Geoscience and Remote sensing*, vol. 50, no. 10, 2020.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

[8] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint:1411.1784*, 2014.

[9] P. Isola, T. Zhu, J.-H.and Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, pp. 5967–5976, 2017.

[10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CVPR*, pp. 105–114, 2017.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[12] C. Fabbri, "Conditional wasserstein generative adversarial networks," *Preprint to be searched in a search engine*, 2016.

[13] P. Wang, S. Ray, C. Peng, Y. Li, and G. Poole, "Premigration deghosting for marine streamer data using a bootstrap approach in tau-p domain," *SEG 83th annual meeting*, 2013.

[14] C. Villani, *Topics in optimal transportation*. American Mathematical Society, 2003.

[15] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," *Proceedings of the 34nd International Conference on Machine Learning*, vol. 70, pp. 214–223, 2017.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5769–5779, 2017.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ICLR*, 2016.

[18] O. Ronneberger, P. Fischer, and T. Brox, "Unet: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, pp. 234–241, 2015.

[19] T. Remez, O. Litany, R. Giryes, and A. Bronstein, "Deep class-aware image denoising," *IEEE International Conference on Image Processing*, pp. 1895–1899, 2017.

[20] Y. Le Cun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," *The Connectionist Models Summer School*, vol. 1, pp. 21–28, 1988.

[21] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameters Estimation*. Society for Industrial and Applied Mathematics (Philadelphia), 2005.

[22] W. Rudin, *Analyse réelle et complexe : cours et exercices*. Dunod, Paris, 1998.

Appendix

A Cross-entropy and gaussian

Training can aim at optimizing a parameterized joint pdf $P_{Y,X}^{(par)} = P_X P_{Y|X}^{(par)}$ to mimic $P_{Y,X}$ the best from a given similarity measure or loss point of view. Cross-entropy represents a common loss choice to minimize [11]:

$$XE(P_{Y,X}, P_{Y,X}^{(par)}) \quad = \quad -\mathbb{E}_{(Y,X)\sim P_{Y,X}} \ln \Big(P_X(X) P_{Y|X}^{(par)}(Y|X) \Big). \tag{14}$$

Dealing with a very general parameterization for $P_{Y|X}^{(par)}$ is unfeasible when $\mathcal{Y}$ has a large dimensionality. However, a simple generalized gaussian parameterization may be considered (the normalization factor is implicit) [21]

$$P_{Y|X}^{(par)}(Y|X) \propto \exp \Big(- ||Y - G_\theta(X)||_{L_p}^p \Big). \tag{15}$$

When eq. (15) is inserted into eq. (14) and the terms that do not contribute to the G_θ optimization are omitted, we obtain eq. (1). Note that variance-like weights can be added to the distance to control the width of the gaussian but they do not affect considerations on the maximum-likelihood of the gaussian.

B Wasserstein distance between two joint pdfs with same marginal

We consider pdfs $P_{Y,X}^i$ (indexed by i) with same marginals P_X, i.e. $P_{Y,X}^i = P_{Y|X}^i P_X$.

$$P_{Y|X}^i(\tilde{Y}|\tilde{X}) = \frac{P_{Y,X}^i(\tilde{Y}, \tilde{X})}{P_X(\tilde{X})} \mathbf{1}_{P_X(\tilde{X}) \neq 0}, \tag{16}$$

where the indicator function constrains the support in $\mathcal{X}$ of $P_{Y|X}^i(\tilde{Y}|\tilde{X})$ to be the same as the support of P_X.

We here use the notations of §4.1. $W_{L_r}(P_{Y|X}^1, P_{Y|X}^2)$, eq. (4), represents a Wasserstein distance between the conditional pdfs $P_{Y|X}^1$ and $P_{Y|X}^2$ for each "parameter" X. Knowing this, we check if

$$JW_{L_r}(P_{Y,X}^1, P_{Y,X}^2) \quad = \quad \mathbb{E}_{X\sim P_X} W_{L_r}(P_{Y|X}^1, P_{Y|X}^2) \tag{17}$$

represents a distance between the joint pdfs $P_{Y,X}^1$ and $P_{Y,X}^2$, i.e. if it satisfies the symmetry and separation properties, and the triangle inequality [22].

JW_{L_r} obviously satisfies the symmetry property: $JW_{L_r}(P_{Y,X}^1, P_{Y,X}^2) = JW_{L_r}(P_{Y,X}^2, P_{Y,X}^1)$.

It also satisfies the separation property

$$P_{Y,X}^1 = P_{Y,X}^2 \Rightarrow P_{Y|X}^1 = P_{Y|X}^2 \Rightarrow W_{L_r}(P_{Y|X}^1, P_{Y|X}^2) = 0 \Rightarrow JW_{L_r}(P_{Y,X}^1, P_{Y,X}^2) = 0$$
$$JW_{L_r}(P_{Y,X}^1, P_{Y,X}^2) = 0 \Rightarrow W_{L_r}(P_{Y|X}^1, P_{Y|X}^2) = 0 \Rightarrow P_{Y|X}^1 = P_{Y|X}^2 \Rightarrow P_{Y,X}^1 = P_{Y,X}^2.$$

The only subtlety lies in the first implication of the first line and the first implication of the second line; they are true because $P_{Y,X}^1, P_{Y|X}^2$ and P_X all have the same support as mentioned above after eq. (16).

Finally, let us check the triangle inequality. As W_{L_r} represents a distance, we have $W_{L_r}(P_{Y|X}^1, P_{Y|X}^3) \leq W_{L_r}(P_{Y|X}^1, P_{Y|X}^2) + W_{L_r}(P_{Y|X}^2, P_{Y|X}^3)$. Taking the P_X expected value straightforwardly leads to

$$JW_{L_r}(P_{Y,X}^1, P_{Y,X}^3) \quad \leq \quad JW_{L_r}(P_{Y,X}^1, P_{Y,X}^2) + JW_{L_r}(P_{Y,X}^2, P_{Y,X}^3).$$

Thus JW_{L_r} defines a distance between two joint pdfs with same marginal with respect to the second parameter.

C Lipschitz dual norm representation

We consider any norm $||.||_d$ and any function $D(\tilde{Y})$ that is derivable with respect to $\tilde{Y}$. The Lipschitz norm $||D||_{Lip_d} = \sup_{\tilde{Y}\neq\tilde{Y}^{(2)}} \frac{|D(\tilde{Y})-D(\tilde{Y}^{(2)})|}{d(\tilde{Y},\tilde{Y}^{(2)})}$, $\forall(\tilde{Y},\tilde{Y}^{(2)}) \in \mathcal{Y}\times\mathcal{Y}$, can then be rewritten

$$||D||_{Lip_d} = \sup_{\tilde{Y}} \sup_{\tilde{Y}^{(2)}\neq 0} \frac{\left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}.\tilde{Y}^{(2)}\right|}{||\tilde{Y}^{(2)}||_d} = \sup_{\tilde{Y}} \sup_{||\tilde{Y}^{(2)}||_d\leq 1} \left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}.\tilde{Y}^{(2)}\right|. = \left|\left|\Phi_{\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}}\right|\right|' \tag{18}$$

In the spirit of Riesz representation theorem related considerations [22], we introduce the function $\Phi_{\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}}(\tilde{Y}^{(2)}) = \frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}.\tilde{Y}^{(2)}$ that is "represented" by $\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}$. If the derivative of $D(\tilde{Y})$ with respect to $\tilde{Y}$ is continuous, we have $\Phi_{\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}} \in \mathcal{Y}'$, where $\mathcal{Y}' = \mathcal{L}(\mathcal{Y},\mathbb{R})$ denotes the topological dual of $\mathcal{Y}$, i.e. the ensemble of the continuous linear functions from $\mathcal{Y}$ into $\mathbb{R}$ [22]. As a consequence, by definition of the dual norm [22], we have

$$\sup_{||\tilde{Y}^{(2)}||_d\leq 1} \left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}.\tilde{Y}^{(2)}\right| = \sup_{||\tilde{Y}^{(2)}||_d\leq 1} \left|\Phi_{\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}}(\tilde{Y}^{(2)})\right| = \left|\left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}\right|\right|_{d'}, \tag{19}$$

where $||.||_{d'}$ represents the dual norm of $||.||_d$. Eqs. (18) and (19) imply

$$||D||_{Lip_d} = \sup_{\tilde{Y}} \left|\left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}\right|\right|_{d'}. \tag{20}$$

The most straightforward choice for the norm d is the L_p norm in the image space, eq. (2), leading to

$$||.||_d = ||\frac{1}{\sigma^p}.||_p \quad \Rightarrow \quad ||.||_{d'} = ||\sigma^q.||_q \quad \text{with} \quad 1/p + 1/q = 1 \tag{21}$$
$$\Rightarrow ||D||_{Lip_{L_p}} = \sup_{\tilde{Y}\in\mathcal{Y}} \Big(\int_\Omega \sigma(y)^q \Big|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}(y)}\Big|^q d\mu(y)\Big)^{\frac{1}{q}}.$$

So, if the L_p norm is chosen in the image space $\mathcal{Y}$, the associated Lipschitz norm will be defined by eq. (21). In our applications, $\sigma(y) = 1$ is taken (we introduced it in eq. (21) for completeness).