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

Thibault Lesieur, Jérémie Messud, Issa Hammoud, Hanyuan Peng, Céline Lacombe, Paulien Jeunesse
CGG, Subsurface Imaging R&D, Massy (France)

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 an input. Surprisingly, our CGAN-based tests on some 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 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 an 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 an 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 in the deterministic prediction case. Then, we point out from the Wasserstein point of view what CGAN

1st I Can't Believe It's Not Better Workshop (ICBINB@NeurIPS 2020), Vancouver, Canada.

should bring compared to an L_p loss, taking the opportunity to discuss the Wasserstein CGAN (W-CGAN) foundations. Our analysis gives a first 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.

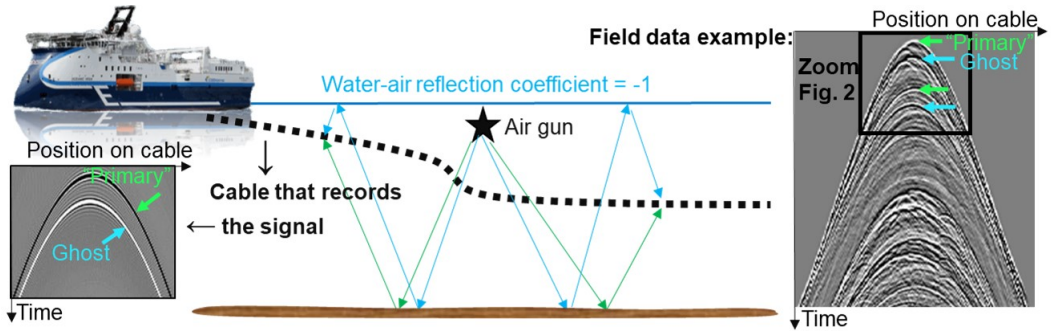

Figure 1: Marine seismic data acquisition. The pressure wavefield generated by an airgun is reflected in the subsurface, then comes back to the surface and is recorded along a cable pulled by a boat. Billions of data over thousands of square kilometers are recorded. A particularity of the geophysical 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.

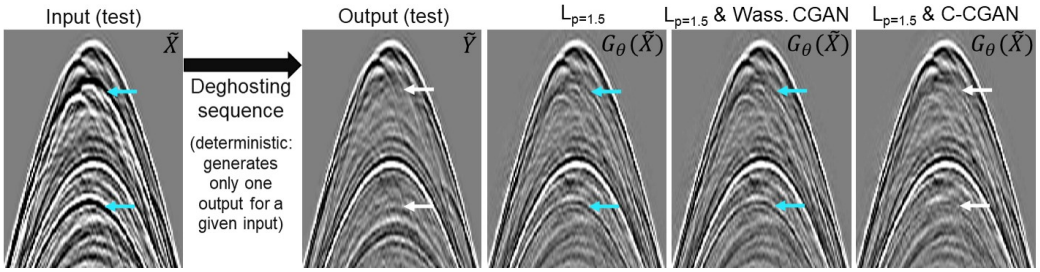

Figure 2: The DNN training data consists in 200 randomly extracted input "images" of size $564 \times 551 \times 1$, representing 0.001% of the total field data, together with corresponding output images generated by a conventional deghosting sequence. On the left, a conventional deghosting result is 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 training). Ghost residuals (blue arrows) can be observed using an L_p loss and adding W-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 seems here 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 (image) space and $\mathcal{Y}$ the output data (image) 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 $P_{Y,X}(\tilde{Y},\tilde{X}) \rightarrow \frac{1}{N_D}\sum_{i=1}^{N_D}\delta(\tilde{Y}-\tilde{Y}_i)\delta(\tilde{X}-\tilde{X}_i)$. $\{\tilde{X}_i,\tilde{Y}_i; i=1..N_D\}$ 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 an L_p loss?

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

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

where the output image space L_p norm is defined by

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

$\mathcal{Y}$ here represents the real $L^p(\Omega)$ space (functions with integrable moments of order p). Each $\tilde{Y} \in \mathcal{Y}$ represents an image indexed by the positions y in a "pixels space" Ω that is measurable for the measure $\mu(y)$. Ω usually represents the (discrete) pixels grid space and μ the counting measure. However, our considerations generalize to continuous spaces Ω taking the Lebesgue measure for μ.

Training aims to minimize $C^p(G_\theta^Z)$, eq. (1), with respect to θ. As C^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 (see Ref. [11] section 6.2.1.2).

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. Of course, an 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 an 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 parameterization like the Gaussian one. XE is not adapted and Wasserstein distances [14] seem like a natural choice. We propose the following Wasserstein-based formulation suited to joint pdfs with same marginal ($p \geq 1$ and $r \geq 1$):

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

$P_{Y|X}$ and $P_{Y|X}^{(par)}$ are considered as single parameter pdfs for each realization of X, and W_{L_p} represents a r-Wasserstein distance between $P_{Y|X}$ and $P_{Y|X}^{(par)}$ for any L_p-norm choice in the output image space [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_p}, 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 $r = 1$ allows to simplify the second line of eq. (4) into the Kantorovitch-Rubinstein (KR) formulation (see [14] section 1.2 and [15])

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

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

$$||D_X||_{Lip_{L_p}} = \sup_{\tilde{Y}\in\mathcal{Y}} \Big|\Big| \frac{\partial D_X(\tilde{Y})}{\partial \tilde{Y}} \Big|\Big|_{L_q} \quad \text{with} \quad 1/p + 1/q = 1. \tag{6}$$

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

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

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

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

- The discriminator's dependency on the input data can possibly be strong and discontinuous, leading in the 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, do not vary rapidly). However, the considerations in this section lead to some clarification on the possibility of using a different discriminator architecture per group of input data with similar properties within W-CGAN if needed.
- Eq. (6) provides the generalization to $L_{p\neq 2}$-norms to the derivative-based Lipschitz constraint of Ref. [16].
- We established that JW_{L_p} represents a distance between two joint pdfs with same marginal.
- Appendix D establishes the link with the scheme obtained starting from a Wasserstein distance between joint pdfs without same marginals [17].

We mentioned in §3 that training with an 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_p}, 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_p} would bring compared to an L_p loss is unclear. This is what we discuss now.

4.2 Advantages of Wasserstein CGAN in the deterministic case?

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

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

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, 18] 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]

$$\widehat{JW}_{L_p}(G_\theta) = \sum_{i=1}^{N_D} \sup_{||\alpha(\tilde{X}_i)||_{L_q}=1} \int_\Omega |\alpha(\tilde{X}_i)(y)| \times \Big| \tilde{Y}_i(y) - G_\theta(\tilde{X}_i)(y) \Big| d\mu(y). \tag{10}$$

In the deterministic prediction limit, the L_p-based loss defined by eqs. (1) and (2) becomes

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

Compared to C^1, i.e. eq. (11) with $p = 1$, we observe that $\widehat{JW}_{L_p}$ adds learnt positive weights $|\alpha(\tilde{X}_i)|$ with unit L_q norm. In other words, with a linear parameterization for F in the deterministic case, W-CGAN "only" adds automatic learning of optimal data-dependent variance-like weights compared to the L_1 loss. In this simple case, these weights can be demonstrated to lead to[2]: $\widehat{JW}_{L_p} = (C^p)^{1/p}$, i.e. to a $\widehat{JW}_{L_p}$ that is equivalent to the L_p-based loss. The takaway is that W-CGAN, eq. (8), should "at least" learn an L_p-based loss or reweighting.

What about more involved parameterizations for F in eq. (9), using for instance 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 postprocessing of the outputs. The supremum principle in eq. (8) allows to learn the postprocessing that makes JW_{L_p} the most sensitive to the differences between the prediction and the output data, i.e. that should concentrate on the less matched events. In configurations where adding such a postprocessing would not affect the relative "positions" of most of the minimums in the loss valley, the main effect should be to improve the training convergence and deal better with the amplitude and noise present in the output data. These situations should tend to occur amongst others when the postprocessings do not dramatically affect the gross data amplitudes hierarchy. This is a first element to interpret when W-CGAN trainings on deterministic geophysical processing sequences may not help to produce a real improvement. The case of Fig. 2 possibly falls in this category, that will be further discussed in §4.4.

[1] We have $JW_{L_p} \rightarrow \sum_{i=1}^{N_D} \sup_{||\alpha(\tilde{X}_i)||_{L_q} \leq 1} \int_\Omega \alpha(\tilde{X}_i)(y) \Big(\tilde{Y}_i(y) - G_\theta(\tilde{X}_i)(y) \Big) d\mu(y)$, 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.

[2] By definition of the "dual norm" ([19] chapter I, [20] chapter IV), applied to the first equation in footnote 1.

Another element is that the method contains free parameters. Two of these are related to the Lipschitz constraint: q (or p) in eq. (6), for which $q \approx 1$ represented a good value, and a weight to impose the constraint using for instance the method of Ref. [16]. Also, like in Refs. [10, 4, 6], we observed JW_{L_p} has to be combined to an $L_{p'}$ loss to give correct results, thus an additional weight is needed. We chose $p' = 1.5$ and tuned the weight so that JW_{L_p} and $L_{1.5}$ losses contribute equally, to obtain the results in Fig. 2. As these hyper-parameters are data dependent, it may explain why W-CGAN did not produce a systematic improvement in our tests. The question of tuning the 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 W-CGAN is that it is not feasible to resolve exactly the supremum principle in JW_{L_p}, eq. (8), at each iteration. This can lead to slowness in the training and possibly sometimes to "localized" unstabilities, that may contribute to the explanation of some poor results. 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

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

where $D_{\tilde{X}_i}$ is defined through eqs. (9) and (12). Eq. (13) looks like an 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 "postprocessed" (through the DNN F) and the corresponding similarly postprocessed 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 W-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 [21]. The F function architecture, that defines the discriminator through eq. (9), is Denet inspired [22]. 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 $q \approx 1$ in the Lipschitz norm, eq. (6), and combined W-CGAN (or C-CGAN) to the $L_{1.5}$ loss, so that both contribute equally. Fig. 2 shows a result at 40 epochs, before full convergence (150 epochs). C-CGAN gave better results than CGAN or $L_{1.5}$ only, on our data. However, the main benefit of C-CGAN is here to accelerate the training. At full convergence we observed the differences between C-CGAN and $L_{1.5}$ become quite smaller, possibly 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 $\mathcal{Y}$. The figure 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 [23]. The adjoint-input is also defined in the $\mathcal{Y}$ space and allows for a visualization of the areas where the events should be better predicted after the update. Fig. 3 shows that the $L_{1.5}$ loss 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 W-CGAN adjoint-input also tends to concentrate on many areas, contributing to explain why it brought no improvement in

Fig. 2. The texture of the latter adjoint-input may seem atypical; we verified that the optimization of eq. (8) and of G_θ converged without unstability, but further analysis regarding the hyper-parameters mentioned in §4.2 is on the way. The C-CGAN adjoint-input, however, learned to concentrate more specifically around the ghost areas, contributing to explain why it converges more rapidly towards an acceptable solution in Fig. 2. Note, however, that the C-CGAN adjoint-input does not exhibit a very strong change in the amplitudes hierarchy compared to the $L_{1.5}$ adjoint-input, contributing to explain that the main effect of C-CGAN would here be to improve the training convergence (remind §4.2).

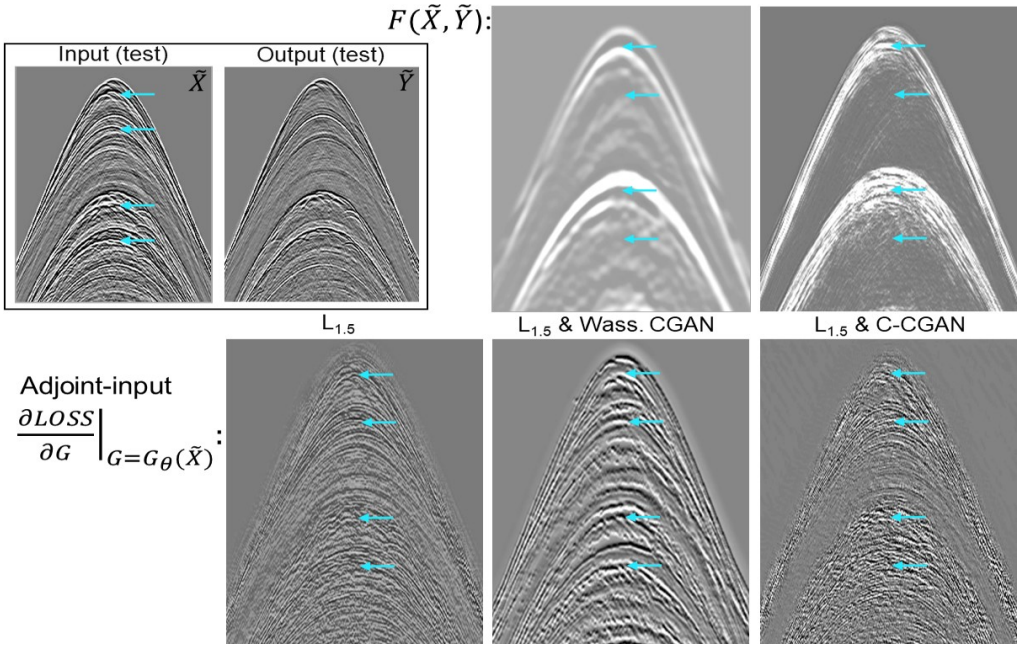

Figure 3: Deghosting task and same data than in Fig. 2. This figure illustrates what the discriminator learns from the $F(\tilde{X},\tilde{Y})$ (top) and adjoint-input (bottom) point of views. We observe that W-CGAN and $L_{1.5}$ tend to "put a weight" on many areas, regardless of their relative importance, 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

We proposed a theoretical analysis of the CGAN framework for predictive tasks. We took the opportunity to establish the CGAN foundations from the Wasserstein point of view, and pointed out what CGAN should bring compared to an L_p loss in the deterministic prediction case. We discussed that W-CGAN may perform more poorly than expected when the corresponding data-space "postprocessings" would not affect the relative "positions" of most of the minimums in the loss valley (for instance when they do not dramatically affect the gross data amplitudes hierarchy), or due to a difficulty with automatically tuning the corresponding hyper-parameters. Another difficulty is that the W-CGAN loss represents a distance only if the supremum principle is perfectly resolved numerically; our C-CGAN formalism helps to keep positivity and gave 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.

Acknowledgments and Disclosure of Funding

The authors are grateful to CGG and Lundin for the permission to publish this work. The authors are indebted to Nicolas Salaun, Samuel Gray, Thibaut Allemand, Gilles Lambaré, Mathieu Chambefort and Stephan Clémençon for enlightening discussions and collaboration.

All funding of this work by CGG.

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] M. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3733–3742, 2017.

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

[19] H. Brezis, *Analyse fonctionnelle: Théorie et applications*. Dunod, Paris, 2020.

[20] W. Rudin, *Functional analysis (second edition)*. Dunod, Paris, 1991.

[21] 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.

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

[23] 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.

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

Appendix

A Cross-entropy and Gaussian

Training can aim at optimizing a parameterized joint pdf $P^{(par)}_{Y,X} = P_X P^{(par)}_{Y|X}$ 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, Ref. [11] section 6.2.1.2:

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

Dealing with a very general parameterization for $P^{(par)}_{Y|X}$ is unfeasible when $\mathcal{Y}$ has a large dimensionality. However, a simple generalized Gaussian parameterization may be considered (the normalization factor is implicit), see Ref. [24] section 4.3:

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

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 "standard-deviation-like" weights $\sigma(y)$, with $\infty > 1/\sigma(y) > 0$, can be added to the distance to control the width of the Gaussian, i.e. $||.||_{L_p} \to ||\frac{1}{\sigma}.||_{L_p}$. 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 $P^i_{Y,X}$ (indexed by i) with same marginal P_X, i.e. $P^i_{Y,X} = P^i_{Y|X} P_X$.

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

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

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

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

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

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

It also satisfies the separation property

$$P^1_{Y,X} = P^2_{Y,X} \Rightarrow P^1_{Y|X} = P^2_{Y|X} \Rightarrow W_{L_p}(P^1_{Y|X}, P^2_{Y|X}) = 0 \Rightarrow JW_{L_p}(P^1_{Y,X}, P^2_{Y,X}) = 0$$
$$JW_{L_p}(P^1_{Y,X}, P^2_{Y,X}) = 0 \Rightarrow W_{L_p}(P^1_{Y|X}, P^2_{Y|X}) = 0 \Rightarrow P^1_{Y|X} = P^2_{Y|X} \Rightarrow P^1_{Y,X} = P^2_{Y,X}.$$

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

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

$$JW_{L_p}(P^1_{Y,X}, P^3_{Y,X}) \quad \leq \quad JW_{L_p}(P^1_{Y,X}, P^2_{Y,X}) + JW_{L_p}(P^2_{Y,X}, P^3_{Y,X}).$$

Thus JW_{L_p} defines a distance between two joint pdfs with same marginal.

C Lipschitz norm dual representation

We consider a real Banach space $\mathcal{Y}$ indexed by the positions y in a set Ω that is measurable for the measure $\mu(y)$. We denote by $||.||_d$ the corresponding norm, not necessarily an L_p norm.

Considering a function $D : \mathcal{Y} \to \mathbb{R}$, the Lipschitz semi-norm is defined by

$$||D||_{Lip_d} = \sup_{\tilde{Y} \neq \tilde{Y}^{(2)}} \frac{|D(\tilde{Y}) - D(\tilde{Y}^{(2)})|}{||\tilde{Y} - \tilde{Y}^{(2)}||_d}, \quad \forall (\tilde{Y}, \tilde{Y}^{(2)}) \in \mathcal{Y} \times \mathcal{Y}. \tag{18}$$

Identifying constant functions, we obtain a norm; this is implicit in the article, hence the use of "Lipschitz norm" throughout the text.

Imposing Lipschitzity through eq. (18) is unpractical numerically, as a global property of D is considered. We would like to turn this global property into a local neighborhood one. Taking $\tilde{Y}^{(2)} = \tilde{Y} + \tilde{\Upsilon}$ with $\tilde{\Upsilon} \in \mathcal{Y}$ and choosing a $D(\tilde{Y})$ that is derivable with respect to $\tilde{Y}$, the Lipschitz norm can be rewritten[3]

$$\begin{aligned} ||D||_{Lip_d} &= \sup_{\tilde{Y}} \sup_{\tilde{\Upsilon} \neq 0} \frac{|D(\tilde{Y} + \tilde{\Upsilon}) - D(\tilde{Y})|}{||\tilde{\Upsilon}||_d} = \sup_{\tilde{Y}} \sup_{\tilde{\Upsilon} \neq 0} \frac{\left|\left(\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}, \tilde{\Upsilon}\right)_\Omega\right|}{||\tilde{\Upsilon}||_d} \\ &= \sup_{\tilde{Y}} \sup_{||\tilde{\Upsilon}||_d \leq 1} \left|\left(\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}, \tilde{\Upsilon}\right)_\Omega\right|, \end{aligned} \tag{19}$$

where the differential is defined through

$$\left(\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}, \tilde{\Upsilon}\right)_\Omega = \int_\Omega \frac{\partial D(\tilde{Y})}{\partial \tilde{Y}(y)} \tilde{\Upsilon}(y) d\mu(y). \tag{20}$$

In the following, $\mathcal{Y}' = \mathcal{L}(\mathcal{Y}, \mathbb{R})$ denotes the topological dual of $\mathcal{Y}$, i.e. the ensemble of the continuous linear functions from $\mathcal{Y}$ into $\mathbb{R}$ ($'$ being the symbol of the topological dual). Note that the differential takes the form in eq. (20) when the Riesz representation theorem ([19] chapter IV) can be invoked in $\mathcal{Y}'$.

We introduce the function $\Phi_{\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}} \in \mathcal{Y}'$ that is "represented" by $\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}$, i.e. $\Phi_{\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}}(\tilde{\Upsilon}) = \left(\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}, \tilde{\Upsilon}\right)_\Omega$. By definition of the "dual norm" ([19] chapter I, [20] chapter IV), we have $\left|\left|\Phi_{\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}}\right|\right|' = \sup_{||\tilde{\Upsilon}||_d \leq 1} \left|\left(\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}, \tilde{\Upsilon}\right)_\Omega\right| = \left|\left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}\right|\right|_{d'}$. In other terms, if the Riesz representation theorem can be invoked in $\mathcal{Y}'$ ([19] chapter IV), the dual norm can be "represented" by a norm $||.||_{d'}$ on $\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}$. Combining these results with eq. (19) gives:

$$\text{If a Riesz representation theorem exists for } \mathcal{Y}': \quad ||D||_{Lip_d} = \sup_{\tilde{Y}} \left|\left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}}\right|\right|_{d'}. \tag{21}$$

Eq. (21) is true for any norm in finite dimensional Banach spaces, that is the case when Ω represents a finite pixels grid and $\mu(y)$ the counting measure. Then, a straightforward choice for the norm d is the L_p norm, eq. (2), for which we have ([19] chapter IV)

$$||.||_d = ||\frac{1}{\sigma}.||_{L_p} \quad \Rightarrow \quad ||.||_{d'} = ||\sigma.||_{L_q} \quad \text{with} \quad 1/p + 1/q = 1, \quad p \geq 1. \tag{22}$$

For completeness, we introduced "standard-deviation-like" weights $\sigma(y)$, that must satisfy $\infty > 1/\sigma(y) > \epsilon$ where $\epsilon > 0$, even if in our applications $\sigma(y) = 1$ is taken. Inserting eq. (22) in eq. (21) leads to

$$\begin{aligned} ||D||_{Lip_{L_p}} &= \sup_{\tilde{Y} \in \mathcal{Y}} \left(\int_\Omega \sigma(y)^q \left|\frac{\partial D(\tilde{Y})}{\partial \tilde{Y}(y)}\right|^q d\mu(y) \right)^{\frac{1}{q}} \\ &= \sup_{\tilde{Y} \in \mathcal{Y}} \left|\left|\sigma \frac{\partial D_X(\tilde{Y})}{\partial \tilde{Y}}\right|\right|_{L_q} \quad \text{with} \quad 1/p + 1/q = 1, \quad p \geq 1. \end{aligned} \tag{23}$$

[3] The second equality is obtained by proving two inequalities. The first inequality involves computing the derivative of $D(\tilde{Y})$ around $\tilde{Y}$. The second inequality involves integrating the derivative of $D(\tilde{Y})$ from $\tilde{Y}$ to $\tilde{Y}^{(2)}$.

So, if $\mathcal{Y}$ represents a finite dimensional $L^p(\Omega)$ space, the "naturally" associated Lipschitz norm is defined by eq. (23) (even if all norms are equivalent in finite dimensional spaces, the "naturally" associated dual norm may lead to better numerical convergence properties...).

What if $\mathcal{Y}$ represents an infinite dimensional $L^p(\Omega)$ space (that is the case for instance when Ω represents a continuous set and $\mu(y)$ the Lebesgue measure)? Eqs. (22)-(23) remain valid but only for $\infty > p \geq 1$. Indeed, no Riesz representation theorem then exists for $\mathcal{Y}'$ when $\mathcal{Y} = L^\infty(\Omega)$.

D General formulation of the Wasserstein distance between two joint pdfs

We here consider two joint pdfs $P_{Y,X} = P_X P_{Y|X}$ and $P^{(2)}_{Y,X} = P^{(2)}_X P^{(2)}_{Y|X}$ that do not necessarilly have the same marginals, i.e. $P_X \neq P^{(2)}_X$ in the general case. We consider the $\mathcal{Y} \times \mathcal{X}$ space to be a Banach space and denote the corresponding norm by $||(\tilde{Y}, \tilde{X}) - (\tilde{Y}^{(2)}, \tilde{X}^{(2)})||$, $\forall((\tilde{Y}, \tilde{X}), (\tilde{Y}^{(2)}, \tilde{X}^{(2)})) \in (\mathcal{Y} \times \mathcal{X}) \times (\mathcal{Y} \times \mathcal{X})$.

A r-Wasserstein distance between $P_{Y,X}$ and $P^{(2)}_{Y,X}$ can be defined by ($r \geq 1$) [17]

$$JW2(P_{Y,X}, P^{(2)}_{Y,X}) = \Big(\inf_{\Pi_{(Y,X),(Y^{(2)},X^{(2)})}} \mathbb{E}_{((Y,X),(Y^{(2)},X^{(2)}))\sim\Pi_{(Y,X),(Y^{(2)},X^{(2)})}} ||(Y,X) - (Y^{(2)}, X^{(2)})||^r \Big)^{\frac{1}{r}}, \tag{24}$$

where the infimum is taken over all joint pdfs $\Pi_{(Y,X),(Y^{(2)},X^{(2)})}$ with marginals $P_{Y,X}$ and $P^{(2)}_{Y,X}$. Switching to the dual formulation for $r = 1$, in the same spirit than in §4.1, leads to

$$JW2(P_{Y,X}, P^{(2)}_{Y,X}) = \sup_{||D||_{Lip2} \leq 1} \Big[\mathbb{E}_{(X,Y)\sim P_{Y,X}} D(Y,X) - \mathbb{E}_{(X^{(2)},Y^{(2)})\sim P^{(2)}_{Y,X}} D(Y^{(2)}, X^{(2)}) \Big], \tag{25}$$

where the discrimator $D : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ is constrained to be 1-Lipschitz for the Lipschitz norm:

$$||D||_{Lip2} = \sup_{(\tilde{Y},\tilde{X}) \neq (\tilde{Y}^{(2)},\tilde{X}^{(2)})} \frac{|D(\tilde{Y},\tilde{X}) - D(\tilde{Y}^{(2)},\tilde{X}^{(2)})|}{||(\tilde{Y},\tilde{X}) - (\tilde{Y}^{(2)},\tilde{X}^{(2)})||}. \tag{26}$$

Now, we consider $P_{Y,X}$ and $P^{(2)}_{Y,X}$ that have the same marginal, i.e. $P_X = P^{(2)}_X$. Eq. (25) becomes

$$JW2(P_{Y,X}, P^{(2)}_{Y,X}) \to \sup_{||D||_{Lip2} \leq 1} \mathbb{E}_{X\sim P_X} \Big[\mathbb{E}_{Y\sim P_{Y|X}} D(Y,X) - \mathbb{E}_{Y^{(2)}\sim P^{(2)}_{Y|X}} D(Y^{(2)}, X) \Big]. \tag{27}$$

In order to establish a relation between eq. (7), JW_{L_p}, and eq. (27), $JW2$, we specify a form for the norm on (X, Y). We consider $\mathcal{Y}$ to be the Banach space used throughout this article (i.e. equipped with the L_p-norm) and $\mathcal{X}$ to be a Banach space (possibly indexed in another space than Ω) equipped with a $||\frac{1}{\sigma}.||_{L_s}$ norm that includes a "standard-deviation-like" weight like in eq. (22). We pose

$$\|(\tilde{Y},\tilde{X}) - (\tilde{Y}^{(2)},\tilde{X}^{(2)})\| = \sqrt{\|\tilde{Y} - \tilde{Y}^{(2)}\|^2_{L_p} + \|\frac{1}{\sigma}(\tilde{X} - \tilde{X}^{(2)})\|^2_{L_s}}. \tag{28}$$

With a similar reasonment than in Appendix C and choosing a $D(\tilde{Y},\tilde{X})$ that is derivable with respect to $\tilde{Y}$ and $\tilde{X}$, one can prove that the Lipschitz norm can be represented by[4]

$$||D||_{Lip2} = \sup_{\tilde{Y},\tilde{X}} \sqrt{\Big|\Big|\frac{\partial D(\tilde{Y},\tilde{X})}{\partial \tilde{Y}}\Big|\Big|^2_{L_q} + \Big|\Big|\sigma \frac{\partial D(\tilde{Y},\tilde{X})}{\partial \tilde{X}}\Big|\Big|^2_{L_t}} \quad \text{with} \quad 1/p + 1/q = 1 \quad \text{and} \quad 1/s + 1/t = 1. \tag{29}$$

Taking the limit $\sigma \to 0$ and denoting equivalently $D(\tilde{Y},\tilde{X})$ or $D_{\tilde{X}}(\tilde{Y})$, eq. (29) reduces to eq. (6). Thus, the JW_{L_p} of §4.1 can be considered as a limiting case of the $JW2$ of this section. Indeed, taking $\sigma \to 0$ aims at imposing $\tilde{X} \to \tilde{X}^{(2)}$ in eq. (26).

[4] The only subtlety is the use of an Euclidean norm between the $\tilde{Y}$ and $\tilde{X}$ components. We did this choice to simplify the computation, the dual representation of the Euclidean norm remainaing the Euclidean norm.